

Fakultät für Informatik

Studiengang B.Sc. Wirtschaftsinformatik

Evaluation von Decision Tree und Random Forest
Klassifikatoren in der Finanzdomäne

Bachelor Thesis

von

Felix Schuhbauer

am tt.mm.2020

Datum der Abgabe: tt.mm.jjjj

Erstprüfer: Prof. Dr. Marcel Tilly

Zweitprüfer: Prof. Dr. Andreas Krüger

ERKLÄRUNG

Ich versichere, dass ich diese Arbeit selbständig angefertigt, nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Rosenheim, den tt.mm.jjjj

Felix Schuhbauer

Abstract

Intelligente Computerprogramme wie Deep Blue und AlphaGo demonstrieren den exponentiellen Fortschritt im Bereich des Machine Learnings der letzten Jahrzehnte. Solche Meilensteine geben Anlass, erfolgreiche Lernalgorithmen neben Brettspielen auf weitere Domänen anzuwenden. Diese Arbeit eruiert am Beispiel der Aktientrendklassifikation, inwiefern solche Modelle in der Finanzdomäne sinnvoll anwendbar sind.

Im Mittelpunkt steht die Frage, ob Decision Trees und Random Forests – zwei weitverbreitete und sehr erfolgreiche Klassifikatoren – Aktientrends genauer vorhersagen können, als ein Dummy Klassifikator. Dabei wird die Rolle des Zeithorizonts, über den sich der klassifizierte Trend erstreckt, im Detail betrachtet. Die Arbeit vergleicht die genannten Klassifikatoren über verschiedene Szenarien hinweg. So werden die häufig angewandten, aber selten untersuchten Einflüsse von Feature Extraction und Hyperparameter-Tuning in diesem Anwendungsfäll analysiert. 11 Technologie-Aktien aus dem S&P 500 Index bilden die Datengrundlage der Untersuchungen. Die Güte von Dummy-, Decision Tree- und Random Forest Klassifikatoren in verschiedenen Situationen wird ermittelt, um die Forschungsfragen zu beantworten. Als Gütemaß dienen dabei, durch Time Series Cross-Validation berechnete, F-Maße. Zudem betrachtet die Arbeit die Konfusionsmetriken Precision und Recall.

Es stellt sich heraus, dass Decision Trees und Random Forests den Dummy Klassifikator nur auf dem längsten Horizont (250 Handelstage) signifikant übertreffen. Im Gegensatz zum Dummy, wird den Baum-basierten Klassifikatoren eine höhere Lernfähigkeit nachgewiesen. Dies geschieht durch eine Gegenüberstellung der F-Maße bei zuerst gleichbleibenden, und dann unterschiedlichen Klassen-Häufigkeitsverteilungen. Die höhere Lernfähigkeit ist vor allem auf die flexiblen Feature-Gewichtungen zurückzuführen: Je länger der Zeithorizont, desto stärker gewichteten Decision Tree und Random Forest jene Features, welche einen ebenso langen, historischen Zeitraum abdecken, und vice versa. Der Random Forest wird in mehreren Untersuchungen insofern als robuster identifiziert, als er auf Änderungen seines Umfeldes schwächer reagiert, als der Decision Tree. Das trifft sowohl auf Feature Extraction, das einen positiven Einfluss auf die Baum-basierten Klassifikatoren hat, als auch auf Hyperparameter-Tuning, das einen stark variablen Einfluss hat, zu.

In einer Interpretation der Ergebnisse aus Sicht des Anwenders wird insbesondere dargelegt, bei welchen Design-Entscheidungen dessen Präferenzen maßgeblich sind. Ausgehend von den Erkenntnissen folgen abschließend Vorschläge für anknüpfende Forschung, darunter die Einführung von Kombinationsfeatures für sämtliche Zeithorizonte.

Inhaltsverzeichnis

1 Einleitung	1
1.1 Motivation	1
1.2 Ziele und Hypothesen	4
2 Theoretische Grundlagen	7
2.1 Machine Learning	7
2.2 Klassifikation	9
2.3 Decision Tree Klassifikator	10
2.3.1 Generelle Funktionsweise und Eigenschaften	10
2.3.2 Erstellung eines Decision Trees	12
2.3.3 Grundlegende (Hyper-)Parameter	15
2.3.4 Optimierung von Decision Trees	15
2.4 Ensemble Methoden	16
2.4.1 Die Ensemble-Idee	16
2.4.2 Bagging und Boosting	16
2.4.3 Entscheidungsfindung im Ensemble	17
2.5 Random Forest Klassifikator	18
2.5.1 Generelle Funktionsweise und Eigenschaften	18
2.5.2 Erstellung eines Random Forests	19
2.5.3 Grundlegende (Hyper-)Parameter	20
2.6 Erfolgsmessung von Klassifikatoren	20
2.7 Underfitting, Overfitting und der Curse of Dimensionality	22
2.8 Besonderheiten bei der Klassifikation von Zeitreihen	23
2.9 Random Walk Theorie und Markteffizienzhypothese	24
2.10 Relevante Literatur	25
3 Methodik	29
3.1 Vorgehen	29
3.2 Tool-Stack und Bibliotheken	33
3.3 Datengrundlage	34
4 Ergebnisse	37
4.1 Erfolgsmessung der Klassifikatoren	37
4.1.1 Auswertung der F-Maße	37
4.1.2 Auswertung auf Aktien-Ebene	40
4.1.3 Veränderung der Klassen-Häufigkeitsverteilungen bei variablem Zeit-horizont	42
4.1.4 Vergleich der Lernfähigkeiten der Modelle	43
4.1.5 Precision und Recall	44
4.1.6 Auswertung pro Aktiengruppe	45
4.2 Einflussgrade der Features in Abhängigkeit vom Zeithorizont	49

4.3 Einfluss von Feature Extraction	52
4.4 Optimierung der Hyperparameter	55
4.5 Interpretation der Ergebnisse aus Anwendersicht	58
5 Schluss	61
5.1 Zusammenfassung und Ausblick	61
A Anhang	65
A.1 Aktienverläufe absolut	65
A.2 Auflistung der Features	71
A.3 Klassen-Häufigkeitsverteilungen der Aktien pro Horizont	72
A.4 Auswertungen der Klassifikatoren: F-Maße pro Aktie pro Horizont als Diagramme	77
A.5 Auswertungen der Klassifikatoren: Precision, Recall und F-Maß in Tabellen .	80
A.6 F-Maße pro Aktie ohne Feature Extraction	86
A.7 Einfluss von Feature Extraction pro Aktie	88
A.8 Auswertungen der Klassifikatoren pro Gruppe	88
A.9 Veränderung des F-Maßes durch Hyperparameter-Tuning	96
Literaturverzeichnis	101

Abbildungsverzeichnis

2.1	Beispielhafter Decision Tree für den Anwendungsfall der Kreditvergabe	11
2.2	Schematischer Ablauf einer Time Series Cross-Validation in den ersten vier Iterationen	24
3.1	Vorgehen in der Untersuchung vom Laden des Datensets bis zur Auswertung .	30
3.2	Relative Kursverläufe der Beispieldatensets ausgehend vom 08.02.2013	34
4.1	F-Maße der Klassifikatoren bei zunehmendem Zeithorizont	39
4.2	F-Maße der vier Klassifikatoren (DT, RF, TunedDT, TunedRF) pro Aktie	40
4.3	Zunahme der Ungleichheit der Klassen-Häufigkeiten pro Horizont	42
4.4	Relative Kursentwicklung der Gruppe 1-Aktien (steigend)	46
4.5	Relative Kursentwicklung der Gruppe 2-Aktien (alternierend)	46
4.6	Relative Kursentwicklung der Gruppe 3-Aktie (fallend)	47
4.7	Durchschnittliche F-Maße der vier Baum-basierten Klassifikatoren pro Gruppe	48
4.8	Einflussgrade der Features im Decision Tree Klassifikator pro Horizont	50
4.9	Einflussgrade der Features im Random Forest Klassifikator pro Horizont	50
4.10	F-Maße der Klassifikatoren mit und ohne Feature Extraction	52
4.11	Einfluss von Feature Extraction auf die F-Maße bei steigendem Zeithorizont .	53
4.12	Einfluss von 10TSCV Hyperparameter-Tuning auf Decision Trees	56
4.13	Einfluss von 10TSCV Hyperparameter-Tuning auf Random Forests	56
A.1	AAPL-Aktienkurs zwischen 2013 und 2018	65
A.2	AMZN-Aktienkurs zwischen 2013 und 2018	66
A.3	CSCO-Aktienkurs zwischen 2013 und 2018	66
A.4	GE-Aktienkurs zwischen 2013 und 2018	67
A.5	GOOGL-Aktienkurs zwischen 2013 und 2018	67
A.6	HP-Aktienkurs zwischen 2013 und 2018	68
A.7	IBM-Aktienkurs zwischen 2013 und 2018	68
A.8	INTC-Aktienkurs zwischen 2013 und 2018	69
A.9	MSFT-Aktienkurs zwischen 2013 und 2018	69
A.10	WU-Aktienkurs zwischen 2013 und 2018	70
A.11	XRX-Aktienkurs zwischen 2013 und 2018	70
A.12	AAPL-Klassenverteilungen pro Horizont	72
A.13	AMZN-Klassenverteilungen pro Horizont	72
A.14	CSCO-Klassenverteilungen pro Horizont	73
A.15	GE-Klassenverteilungen pro Horizont	73
A.16	GOOGL-Klassenverteilungen pro Horizont	74
A.17	HP-Klassenverteilungen pro Horizont	74
A.18	IBM-Klassenverteilungen pro Horizont	75
A.19	INTC-Klassenverteilungen pro Horizont	75
A.20	MSFT-Klassenverteilungen pro Horizont	76
A.21	WU-Klassenverteilungen pro Horizont	76

A.22	XRX-Klassenverteilungen pro Horizont	77
A.23	Dumm: F-Maße pro Horizont pro Aktie	77
A.24	Decision Tree: F-Maße pro Horizont pro Aktie	78
A.25	Random Forest: F-Maße pro Horizont pro Aktie	78
A.26	Decision Tree mit Tuning: F-Maße pro Horizont pro Aktie	79
A.27	Random Forest mit Tuning: F-Maße pro Horizont pro Aktie	79
A.28	Gruppe 1: F-Maße des Dummy Klassifikators pro Horizont pro Aktie	88
A.29	Gruppe 1: F-Maße des Decision Trees pro Horizont pro Aktie	89
A.30	Gruppe 1: F-Maße des Random Forests pro Horizont pro Aktie	89
A.31	Gruppe 1: F-Maße des Decision Trees mit Tuning pro Horizont pro Aktie	90
A.32	Gruppe 1: F-Maße des Random Forests mit Tuning pro Horizont pro Aktie	90
A.33	Gruppe 2: F-Maße des Dummy Klassifikators pro Horizont pro Aktie	91
A.34	Gruppe 2: F-Maße des Decision Trees pro Horizont pro Aktie	91
A.35	Gruppe 2: F-Maße des Random Forests pro Horizont pro Aktie	92
A.36	Gruppe 2: F-Maße des Decision Trees mit Tuning pro Horizont pro Aktie	92
A.37	Gruppe 2: F-Maße des Random Forests mit Tuning pro Horizont pro Aktie	93
A.38	Gruppe 3: F-Maße des Dummy Klassifikators pro Horizont pro Aktie	93
A.39	Gruppe 3: F-Maße des Decision Trees pro Horizont pro Aktie	94
A.40	Gruppe 3: F-Maße des Random Forests pro Horizont pro Aktie	94
A.41	Gruppe 3: F-Maße des Decision Trees mit Tuning pro Horizont pro Aktie	95
A.42	Gruppe 3: F-Maße des Random Forests mit Tuning pro Horizont pro Aktie	95
A.43	Einfluss von 5TSCV Hyperparameter-Tuning auf Decision Trees	96
A.44	Einfluss von 5TSCV Hyperparameter-Tuning auf Random Forests	96
A.45	Einfluss von 3TSCV Hyperparameter-Tuning auf Decision Trees	97
A.46	Einfluss von 3TSCV Hyperparameter-Tuning auf Random Forests	97
A.47	Einfluss von 5fCV Hyperparameter-Tuning auf Decision Trees	98
A.48	Einfluss von 5fCV Hyperparameter-Tuning auf Random Forests	98
A.49	Einfluss von 3fCV Hyperparameter-Tuning auf Decision Trees	99
A.50	Einfluss von 3fCV Hyperparameter-Tuning auf Random Forests	99

Tabellenverzeichnis

2.1	(Hyper-)Parameter eines Decision Trees und deren Bedeutung	14
2.2	(Hyper-)Parameter eines Random Forests und deren Bedeutung	20
2.3	Aufbau einer Konfusionsmatrix	21
3.1	Die fünf zentralen Variablen für die Untersuchungen	29
3.2	Beispielwerte für die Features aus dem AAPL-Datenset	35
3.3	Beispielwerte für die Klassen aus dem AAPL-Datenset	35
4.1	Erfolgsmessung pro Klassifikator für die verschiedenen Zeithorizonte	38
4.2	Konfusionsmatrix der vier Klassifikatoren für den 250d-Horizont auf XRX	40
4.3	Konfusionsmatrix der vier Klassifikatoren für den 250d-Horizont auf GE	41
4.4	Einteilung der Aktien in Gruppen zur weiteren Analyse	45
4.5	Summen der 1d/5d- (gelb) bzw. 65d/250d-Features (rot) pro Horizont	51
4.6	Veränderung der F-Maße pro Klassifikator durch Feature Extraction	53
4.7	Veränderung der F-Maße durch Feature Extraction pro Horizont	54
4.8	Veränderung der F-Maße durch Feature Extraction pro Gruppe	55
A.1	Konfusionsmetriken im Durchschnitt über alle Datensets	80
A.2	Konfusionsmetriken auf AAPL	80
A.3	Konfusionsmetriken auf AMZN	81
A.4	Konfusionsmetriken auf CSCO	81
A.5	Konfusionsmetriken auf GE	82
A.6	Konfusionsmetriken auf GOOGL	82
A.7	Konfusionsmetriken auf HP	83
A.8	Konfusionsmetriken auf IBM	83
A.9	Konfusionsmetriken auf INTC	84
A.10	Konfusionsmetriken auf MSFT	84
A.11	Konfusionsmetriken auf WU	85
A.12	Konfusionsmetriken auf XRX	85
A.13	F-Maße pro Aktie ohne Feature Extraction	86
A.14	Einfluss von Feature Extraction pro Aktie	88

1 Einleitung

1.1 Motivation

Als erstes intelligentes Schachprogramm bezwang Deep Blue den amtierenden Schachweltmeister, Garry Kasparov, im Jahr 1997. Diese Partie gilt als Meilenstein der Künstlichen Intelligenz. Das von IBM entwickelte Programm schaffte es, den Zustandsraum des Brettspiels – mit einer Komplexität von 10^{120} (Shannon, 1950) – in regulärer Spielzeit auf Profi-Niveau zu lösen. Bis zu 330 Millionen Spielpositionen kann Deep Blue pro Sekunde auswerten (Campbell u. a., 2002). Eine Maschine hat den Menschen im Schach besiegt. Weltweit wird dieses Spektakel in den Medien verfolgt. So veröffentlichte zum Beispiel in Großbritannien The Guardian den Artikel "Deep Blue win a giant step for computerkind" (Harding u. Barden, 1997), in den USA berichteten unter anderem die New York Times ("Deep, Deeper, Deepest Blue" (Johnson, 1997)) und das Wall Street Journal ("IBM's Winning 'Deep Blue' Is Still Product of Primates" (Ziegler, 1997)). Infolge dieses Schach-Duell hat sich die öffentliche Wahrnehmung von Künstlicher Intelligenz gewendet: von Science Fiction zur Realität.

18 Jahre später folgte ein weiterer Durchbruch. DeepMind's AlphaGo besiegt den damaligen Go-Weltmeister, Lee Sedol (Silver u. a., 2017). Das traditionsreiche asiatische Brettspiel Go, vor mehr als 2.500 Jahren in China erfunden, birgt, ohne die Reihenfolge der Spielzüge zu beachten, 10^{170} mögliche Spielzustände – also einen um den Faktor hundert Oktillionen, 10^{50} , größeren Zustandsraum als Schach. Die Anzahl der möglichen Brettkonstellationen im Go ist somit größer als die Anzahl der Atome im Universum (Silver u. Hassabis, 2016). Nur zwei Jahre später besiegte AlphaGo Zero, der Nachfolger von AlphaGo, diesen mit 100 zu Null.

Während Schach und Go jeweils für zwei Spieler ausgelegt sind und beide Spiele mit vollständigen Informationen sind, ist das Pokerspiel Texas Hold'em deutlich komplexer: Es ist für sechs Spieler ausgelegt, und es ist ein Spiel unter unvollständigen Informationen. Doch Pluribus, ein Programm, das Pokern lernt, indem es gegen sich selbst spielt, bezwang im Jahr 2019 einige der besten Poker-Spieler der Welt (Brown u. Sandholm, 2019). Eine Neuheit war dabei das Erlernen von Bluffing, dem taktischen Täuschen der Mitspieler. Die Künstliche Intelligenz betrachtet nicht mehr ausschließlich Wahrscheinlichkeiten, logische Regeln und mathematische Zusammenhänge. Stattdessen bewertet das Programm das Verhalten seiner Mitspieler und nutzt deren Schwächen aus. Nicht nur im Poker ist Bluffing ein mächtiges Instrument. Dwight D. Eisenhower, von 1953 bis 1961 Präsident der USA, hat so unter anderem durch die angebliche Bereitschaft, Atomwaffen einzusetzen, Kriege vorgebeugt und die Geschichte maßgeblich beeinflusst (Thomas, 2013).

Künstliche Intelligenz hat in jüngerer Zeit regelmäßig scheinbar unlösbare Aufgaben gemeistert. Der exponentielle Fortschritt in den letzten 25 Jahren, oft anhand von Brettspielen veranschaulicht, hat Forscher, Politiker und Unternehmen dazu angeregt, diese Methoden auch in anderen Domänen zu erproben und einzusetzen. Doch die Entwicklung dieses Feldes verlief nicht immer so erfolgreich.

Das Begriff der Künstlichen Intelligenz entstand auf der Dartmouth Konferenz von 1956

(Russell u. Norvig, 2009). Das Ziel dieses interdisziplinären Forschungsgebietes, das mitunter Informatiker, Mathematiker, Neurowissenschaftler und Philosophen beschäftigt, war es "Maschinen zu entwerfen, die Abstraktionen bilden, um damit Probleme zu lösen, die zuvor nur Menschen lösen konnten, und sich selbst verbessern" (McCarthy u. a., 1955). Es folgte großer Enthusiasmus. Der General Problem Solver löste Puzzles und die Türme von Hanoi erstmals auf menschliche Art und Weise (Russell u. Norvig, 2009). Expertensysteme, die dem Nutzer basierend auf gespeicherten Regeln Ratschläge erteilen, trieben die Kommerzialisierung der neuen Systeme an. Im Jahr 1982 hat das Expertensystem R1 der Digital Equipment Corporation geschätzte 40 Millionen US-Dollar jährlich eingespart, indem es die Bestellung von neuen Computern unterstützte (Russell u. Norvig, 2009). Weitere Unternehmen folgten und investierten Milliarden in die Entwicklung intelligenter Programme. Die Erwartungen an die Künstliche Intelligenz nahmen rapide zu. Die Technologie war jedoch noch nicht ausgereift und blieb hinter den Erwartungen zurück. Ab 1988 tritt eine Phase der Ernüchterung ein, der sogenannte "KI-Winter" (Russell u. Norvig, 2009).

Der KI-Winter hielt bis in die 1990er Jahre an, als die steigende Menge an Daten, höhere Rechenleistungen und neue Algorithmen dem Feld wieder Aufschwung verliehen. Der Erfolg von Deep Blue markiert einen Wendepunkt. Spätestens seit der Jahrtausendwende erlebt das Feld eine Renaissance. Um die Erforschung von Künstlicher Intelligenz transparent zu machen und zu analysieren, veröffentlicht das Human-Centered AI Institute (HAI) der Stanford Universität den AI Index. Demnach hat sich die Anzahl wissenschaftlicher Veröffentlichungen auf der Scopus-Datenbank mit KI-Bezug zwischen 1996 und 2017 um das Siebenfache erhöht, während Informatik-Veröffentlichungen um das Fünffache und alle Veröffentlichungen nicht einmal um das Doppelte zulegten (Shoham u. a., 2018).

Die Bedeutung von Künstliche Intelligenz schlägt sich nicht nur in Forschung und Wirtschaft, sondern auch in der Politik nieder. Staaten machen darauf aufmerksam, dass intelligente Programme einen zunehmenden Einfluss auf das Zusammenleben unserer Gesellschaft haben, und versuchen, in diesem Bereich eine Vorreiterrolle einzunehmen. Die Bundesministerin für Bildung und Forschung, Anja Karliczek, hat Künstliche Intelligenz als "den Treiber des Wandels, unseres Zusammenlebens und unserer Arbeit" (Karliczek, 2019) beschrieben und das Wissenschaftsjahr 2019 als das der Künstlichen Intelligenz ausgerufen (BMBF, 2019). So soll der Dialog zwischen Forschung und Gesellschaft gefördert, und Wissenschaft für die Bevölkerung erlebbar gemacht werden.

In der Wirtschaft, und speziell in der Finanzdomäne, ist ein Anwendungsgebiet die Vermögensverwaltung. BlackRock Inc., der weltweit größte Vermögensverwalter, administriert über 6,5 Billionen US-Dollar, und setzt dazu auch Künstliche Intelligenz ein (Szmigiera, 2019). Das Unternehmen hat eine Risiko- und Investmentplattform namens Aladdin entwickelt, die Daten analysiert und aufbereitet. In das Modell fließen über 2.000, täglich berechnete Faktoren ein, wie zum Beispiel Wechselkurse oder Zinssätze (BlackRock Inc., 2020a). Auf die Plattform hat nicht nur BlackRock selbst Zugriff, sondern auch andere Vermögensverwalter wie Versicherer oder Vorsorgeeinrichtungen, die das Programm entgeltlich nutzen können. So werden über Aladdin Anlagen im Wert von über 14 Billionen US-Dollar, für mehr als 160 Kunden, und insgesamt 30.000 Investmentportfolios verwaltet (BlackRock Inc., 2020b).

Intelligente Maschinen machen weniger Fehler, sind schneller und sind, gemessen an Brettspielen, schlauer als Menschen. Aber müssen sie den Menschen deshalb in Zukunft ersetzen? Nicht unbedingt. Bei der Anlageentscheidung, zum Beispiel, wird Künstliche Intelligenz ergänzend eingesetzt und unterstützt den Menschen, anstatt ihn zu ersetzen (Pozen u.

Ruane, 2019). Ein großer Vorteil dabei ist die Objektivität des Computers. Menschen, die an den Aktienmärkten investieren, handeln von Natur aus irrational. Zwei verbreitete kognitive Verzerrungen sind die Verlustaversion, nach der Verluste höher gewichtet werden als Gewinne, und der Bestätigungsfehler, laut dem die Anleger Informationen so interpretieren, dass diese ihre Einschätzungen bestätigen (Pozen u. Ruane, 2019). Programme vermeiden solche kognitiven Verzerrungen, indem sie ausschließlich auf Basis von Daten entscheiden. Dadurch sind sie jedoch anfälliger für neue Verzerrungen, beispielsweise durch ungeeignete Beispieldaten. Ein Beispiel für eine solche komplementäre Beziehung zwischen Mensch und Maschine ist heute im Börsenhandel vorzufinden: Dort stammen 60% der Handelsaktivitäten von Programmen, die von Menschen vorgegeben Regeln anwenden (Economist, 2019).

Aufgrund ihrer hohen Bedeutung für die Gesellschaft, zum Beispiel im Zuge der Altersvorsorge, sowie der monetären Anreize für einzelne Investoren, erfährt die Finanzdomäne eine hohe Aufmerksamkeit in der Erforschung von Künstlicher Intelligenz. Die Finanzdomäne umfasst hier vor allem private und institutionelle Kapitalgeber, darunter Versicherungen, Stiftungen und Pensionskassen, die ihre finanziellen Mittel Kapitalnehmern, häufig Unternehmen oder Staaten, für eine Gegenleistung zur Verfügung stellen. Die Kapitalmärkte sind der Handelsplatz, an dem Kapitalangebot und -nachfrage aufeinandertreffen. Handelt es sich um Eigenkapital, das in Form von Aktien gehandelt wird, so spricht man vom Aktienmarkt. Diese Arbeit fokussiert sich auf den sekundären Aktienmarkt, auf dem die Marktteilnehmer Aktien untereinander handeln. Dieser Handel findet größtenteils an Börsen statt. Anleger müssen sich entscheiden, welche Aktien sie kaufen, und welche sie verkaufen. Ein zentrales Entscheidungskriterium des Anlegers ist die von ihm erwartete Kursentwicklung des Wertpapiers. Der Anleger ist also daran interessiert, wie sich der Trend des Aktienkurses zukünftig verhält: Steigt er? Fällt er? In welchem Zeitraum steigt oder fällt er? Diese Fragen sollen die nachfolgend untersuchten Klassifikatoren beantworten.

Der breite Begriff der Künstlichen Intelligenz kann in vier Kategorien unterteilt werden: menschliches Denken, menschliches Handeln, rationales Denken und rationales Handeln (Russell u. Norvig, 2009). Diese Arbeit beschränkt sich auf einen Teilbereich des menschlichen Handelns, das Machine Learning. Machine Learning ist jener Teilbereich, der sich mit Programmen beschäftigt, die sich an ihre Umwelt anpassen indem sie Muster erkennen und extrapoliieren können (Russell u. Norvig, 2009).

Diese Arbeit untersucht speziell Decision Trees und Random Forests, weil sich diese in der Forschung als sehr zuverlässig herausgestellt haben, vielseitig anwendbar sind, und zudem für den Menschen interpretierbare Machine Learning Modelle sind.

Der Decision Tree ist einer der simpelsten, aber dennoch erfolgreichsten Lernalgorithmen (Russell u. Norvig, 2009). Da seine Ergebnisse für den Menschen nachvollziehbar sind und er die Grundlage für den Random Forest ist, eignet er sich zur Untersuchung, da er so als Vergleichsmaßstab dienen kann.

Random Forests gehören zu den mächtigsten Machine Learning Algorithmen der modernen Zeit (Gron 2017; Biau u. Scornet 2016). Sie haben auch bei hoch-dimensionalen Vorhersagen zuverlässige und stabile Genauigkeiten bewiesen (Tang u. a. 2018; Belgiu u. Drăguț 2016). In verschiedenen Domänen wurde ihre praktische Anwendbarkeit bereits unter Beweis gestellt. So nennen beispielsweise Biau u. Scornet (2016) erfolgreiche Anwendungen in der Chemoinformatik (Svetnik u. a., 2003), Ökologie (Prasad u. a. 2006; Cutler u. a. 2007), 3D-Objekterkennung (Shotton u. a., 2011) und Bioinformatik (Diaz-Uriarte u. Alvarez, 2006). In der Finanzdomäne sind Random Forests prominent vertreten, zum Beispiel zur

Risikoanalyse. Tang u. a. (2018) haben ein Frühwarnsystem für Bankenversagen mithilfe eines Random Forest Klassifikators programmiert. Anhand der Jahresberichte von über 15.000 Banken weltweit hat das Modell Vorhersagen getroffen, wie anfällig diese Banken für Vermögensverluste sind. Ein häufig untersuchtes Anwendungsbereich ist die Vorhersage von Aktienkursen (Lohrmann u. Luukka 2019; Sadia u. a. 2019; Maini u. Govinda 2017; Alavi u. a. 2015). Pasupulety u. a. (2019) haben Random Forests mit Sentiment Analysis kombiniert, um Aktienkurse vorherzusagen.

Ein großer Vorteil von Decision Trees und Random Forests ist, dass sie nicht nur zur Klassifikation selbst, sondern auch zur Erkennung der wichtigsten Features einsetzbar sind. Zum Beispiel haben Cui u. a. (2018), auf Basis amerikanischer 10-Q Quartalsberichte, die Ausgabe weiterer Aktien von öffentlich gelisteten Unternehmen vorhergesagt. Dabei haben sie Random Forests zusätzlich zur Klassifikation auch zur Bestimmung des wichtigsten Features benutzt. Das englische Wort für Fusion, "merger", wurde so als das aussagekräftigste für die sekundäre Aktienausgabe identifiziert.

Aufgrund der vielfältigen Einsatzmöglichkeiten von Decision Trees und Random Forests ist die weitere Erforschung derselben vielversprechend. Erkenntnisse über ihre Eigenschaften und Verhalten können für mehrere Forschungsäste einen Mehrwert stiften.

1.2 Ziele und Hypothesen

Die Forschungsfrage dieser Arbeit ist, ob Decision Tree und Random Forest Klassifikatoren sinnvoll zur Aktientrendklassifikation eingesetzt werden können. Um diese Frage zu beantworten, werden die Treffergenauigkeiten der Modelle mit jener des Dummy Klassifikators verglichen. Übertreffen Decision Tree oder Random Forest den Dummy signifikant, so wird deren Einsatz als sinnvoll beurteilt.

Um die Forschungsfrage gründlich zu untersuchen, bieten sich relevante Teilfragen an. So soll die Arbeit herausfinden, welche Rolle der Zeithorizont der Klassifikation spielt. Die Genauigkeit der Modelle bei 1d könnte im Gegensatz zu Handelsjahr 250d stark abweichen. Die Vermutung liegt nahe, das kurzfristige Trendvorhersagen, aufgrund des Random Walks und der Emotionalität, durchschnittlich ungenauer sind als langfristige, die einen ausgeprägteren Trend verfolgen. Für die weiteren nachfolgenden Teilfragen soll stets zusätzlich eine Differenzierung der Ergebnisse nach Zeithorizont vorgenommen werden. Es ist zum Beispiel vorstellbar, dass die Berechnung neuer Features für unterschiedliche Zeithorizonte zu abweichenden Ergebnissen führt.

Weiterhin soll das die Gewichtung der Features in Abhängigkeit von dem Zeithorizont untersucht werden. Weisen die Baum-basierten Klassifikatoren eine hinreichend hohe Lernfähigkeit auf und haben langfristige Features mehr Aussagekraft für langfristige Prognosen als für kurzfristige, so ist zu erwarten, dass die Modelle die Feature-Gewichtungen entsprechend anpassen. Zur Klassifikation des Trends über das gesamte nächste Jahr sollten dann die längerfristigen Features höher gewichtet werden, als zur Bestimmung des Trends über die nächsten fünf Tage. Unterschiede in der Feature-Gewichtung zwischen dem Random Forest und dem Decision Tree sollen entdeckt und erläutert werden. Der Dummy Klassifikator nimmt keine Gewichtung der Feature vor und kann somit diesbezüglich nicht verglichen werden. Eine weitere Teilfrage besteht in der Erfolgsmessung der Modelle auf den einzelnen Aktien. Es ist anzunehmen, dass sich die Treffergenauigkeiten von Aktie zu Aktie erheblich unterscheiden.

Um herauszufinden, wie die Genauigkeit der Klassifikatoren maximiert werden kann,

finden Feature Extraction und Hyperparameter-Tuning Anwendung. Diese Methoden führen häufig zu einer Verbesserung der Modelle. Denn zum einen stehen mit Feature Extraction mehr Features, und somit mehr und möglicherweise aussagekräftigere Informationen, zur Verfügung. Zum anderen reduziert Hyperparameter-Tuning die Gefahr von Overfitting und damit die Gefahr von niedrigen Trefferquoten auf unbekannten Instanzen. Aus diesen Gründen sollte in dieser Arbeit für beide Ansätze eine Erhöhung der Treffergenauigkeiten zu beobachten sein.

Transaktionskosten, die durch den Aktienkauf oder -verkauf in der Realität entstehen, werden vernachlässigt, da sie von zahlreichen Faktoren abhängen, stark variieren und deshalb nicht sinnvoll berechenbar sind. Auch beschränkt sich die Arbeit auf binäre Klassifikation; es werden ausschließlich die zwei Trendklassen "steigend" und "fallend" betrachtet.

Diese Arbeit stellt zunächst in Abschnitt 2 die grundlegende Theorie dar, auf die sich die Untersuchungen und anschließenden Auswertungen stützen. Ebenso wird auf die aktuelle Literatur und Forschungslücken hingewiesen. Als nächstes erläutert Abschnitt 3 das Vorgehen in den Untersuchungen samt der verwendeten Datensets. Die Beschreibung der Ergebnisse und eine kritische Diskussion dergleichen erfolgt in Abschnitt 4. Abschließend werden die Erkenntnisse zusammengefasst und durch diese hergeleitete Vorschläge für anknüpfende Forschung präsentiert.

2 Theoretische Grundlagen

2.1 Machine Learning

Machine Learning bedeutet, einen Computer so zu programmieren, dass ein bestimmtes Leistungskriterium anhand von Beispieldaten oder Erfahrungswerten aus der Vergangenheit optimiert wird (Alpaydin, 2008). Diese Programme führen für einen gegebenen Input nicht immer zum selben Ergebnis, sondern lernen – ähnlich wie der Mensch – anhand von Beispielen. Deshalb eignet sich Machine Learning insbesondere für Probleme, für die der Mensch keine simplen Regeln verfassen und in einem Algorithmus automatisieren kann. Stattdessen löst der Mensch solche Probleme anhand von Erfahrungen, teilweise anhand von Intuition. Ein Beispiel ist die Diagnose von Krebszellen. Fachärzte durchlaufen eine jahrelange Ausbildung und analysieren eine Vielzahl von Beispielpbildern, um eine Einschätzung über neue Zellen treffen zu können. Eine solche Diagnose kann nicht mit einem simplen Algorithmus automatisiert werden, da jedes Zellenbild einzigartig ist. Es ist nicht realistisch, in einem Algorithmus alle Eventualitäten programmatisch abzudecken. Machine Learning hingegen ermöglicht es, den menschlichen Lernprozess nachzubilden und somit komplexe Diagnosen zu erstellen. Vorteile gegenüber dem Menschen sind dabei die Objektivität, da der Computer keinen Emotionen unterliegt, und die Fähigkeit, große Mengen an Daten in einem Bruchteil der Zeit, die ein Mensch benötigen würde, zu vergleichen.

Machine Learning gliedert sich in die drei Bereiche des Supervised-, Unsupervised- und Reinforcement Learnings. Reinforcement Learning umfasst Modelle, die durch Feedback im Form von Belohnungen und Bestrafungen lernen (Russell u. Norvig, 2009). Beim Unsupervised Learning werden die Modelle mit Daten trainiert, die zuvor nicht mit einer Klasse betitelt wurden. Stattdessen werden Ähnlichkeiten unter den Datensätzen gesucht, um sie zum Beispiel in Cluster einzuteilen. Diese Arbeit beschäftigt sich ausschließlich mit dem dritten Bereich, dem Supervised Learning. Hier wird das Modell mit Datensätzen trainiert, die jeweils sowohl den Input als auch den gewünschten Output, die Klasse, enthalten. Die Aufgabe des Modells ist es anschließend, für unbekannte Datensätze anhand des Inputs deren Klassen zu bestimmen.

Technisch gesehen lernt ein Machine Learning Programm, indem es eine Funktion

$$f(X|\Theta) = \hat{y}, \quad f \in F \tag{2.1}$$

aus dem Funktionenraum F bildet und durch Training mit Beispieldaten die Parameter Θ optimiert. Der Funktionenraum F umfasst alle Funktionen, die ein Modell erlernen kann, und ist vom gewählten Lernalgorithmus abhängig. Die Beispieldaten sind eine Stichprobe χ mit n Datensätzen in der Form

$$\chi = \{(X_1, y_1), \dots, (X_n, y_n)\}. \tag{2.2}$$

Dabei steht X für einen m -dimensionalen Input-Vektor der Form

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \quad (2.3)$$

der als Informationsgrundlage zur Bestimmung von \hat{y} dient. Die Variable \hat{y} bezeichnet das Ergebnis der Funktion für einen gegebenen Input-Vektor X . Die Optimierung von $f(X|\Theta)$ bezieht sich hier auf die Suche jener Parameter Θ , welche die genaueste Näherung für die tatsächliche Funktion $f^*(X)$ und somit auch für die tatsächliche Klasse y erzielen (Alpaydin, 2008). Ist die Variable \hat{y} numerisch, so spricht man von Regression. Ein Beispiel für eine Regression ist die Vorhersage der Temperatur in Grad Celsius. Dabei enthält die Ergebnismenge der Funktion unendlich viele Elemente, zum Beispiel die reellen Zahlen. Ist die Variable \hat{y} kategorisch, so spricht man von Klassifikation. Im Spezialfall einer binären Klassifikation gilt zudem

$$\hat{y} = \begin{cases} 1, & \text{wenn } X \text{ eine positive Instanz ist} \\ 0, & \text{wenn } X \text{ eine negative Instanz ist.} \end{cases} \quad (2.4)$$

Welche Instanzen positiv und welche negativ sind, ist vom Anwender zu definieren. Ein Anwendungsbeispiel für eine binäre Klassifikation ist die Bestimmung, ob eine gegebene Pflanze für den Menschen giftig ist oder nicht. Bei der Klassifikation ist die Menge der möglichen Werte von \hat{y} endlich. In binären Fall kann \hat{y} , wie in Gleichung 2.4 gezeigt, genau zwei Werte annehmen.

Nach dem Training trifft das Modell Aussagen über Instanzen, die außerhalb von χ liegen. Die Generalisierungsfähigkeit des Modells mit Parametern Θ wird anhand der Abweichungen zwischen \hat{y} und y für alle i Instanzen eines Validierungsdatensets χ_{val} mittels einer Fehlerfunktion

$$E(\Theta|\chi_{val}) = \sum_i Diff(\hat{y}_i, y_i) \quad (2.5)$$

berechnet. Für die Diff-Funktion gibt es zahlreiche Methoden, deren Eignung vom konkreten Anwendungsfall abhängt (Alpaydin, 2008). Die in dieser Arbeit verwendeten Methoden sind in Kapitel 3, "Methodik", erläutert. Das finale Modell verwendet anschließend jene Funktion, die die geringsten Abweichungen auf χ_{val} erreicht, also die beste Generalisierung aufweist. Um die erwartete Fehlerrate des finalen Modells zu bestimmen, werden die Abweichungen auf einem Testdatenset χ_{test} herangezogen. Die Aufgabe von Machine Learning ist es also, jene Parameter Θ_{opt} zu finden, welche die Fehlerfunktion minimieren, also

$$\Theta_{opt} = \underset{\Theta}{\operatorname{argmin}} E(\Theta|\chi). \quad (2.6)$$

An dieser Stelle sei auf einen grundlegenden Trade-Off im Machine Learning hingewiesen: die Komplexität der Funktion, bedingt durch die Mächtigkeit des Funktionenraumes F , die Menge an Daten in χ und der Generalisierungsfehler sind voneinander abhängig (Alpaydin, 2008). Steigende Komplexität führt bei gleichbleibender Datenbasis zu einem

höheren Generalisierungsfehler, kann aber bei größerer Datenbasis zu einem niedrigeren Generalisierungsfehler führen. Wenn die Datenmenge *ceteris paribus* steigt, nimmt der Generalisierungsfehler ab, und vice versa (Alpaydin, 2008).

Der Schwerpunkt dieser Arbeit liegt auf der Klassifikation. Deshalb wird diese in der folgenden Sektion genauer betrachtet.

2.2 Klassifikation

Klassifikation bezeichnet das Zuweisen einer Klasse \hat{y} zu einem gegebenen Input-Vektor X . Ein Beispiel ist die Bestimmung der Kreditwürdigkeit eines Bankkunden, bekannt als das Credit Scoring Problem (Abdou u. Pointon, 2011). Die Bank hat bei der Vergabe eines Kredites das Ziel, das Ausfallrisiko zu minimieren und den erwarteten Gewinn zu maximieren. Dazu werden von Kredit-Antragsstellern Datenpunkte wie Vermögen, Gehalt, Kredithistorie und Alter erhoben. Die gesammelten Daten dienen anschließend als Trainingsdaten für einen Machine Learning Algorithmus. Dieser erstellt ein Modell, das für die historischen Daten optimiert ist. Dieses Modell klassifiziert anschließend die Kreditwürdigkeit neuer Antragssteller. Dadurch wird die Bank bei der Kreditentscheidung unterstützt. In manchen Fällen wird die Entscheidung anhand von Schwellenwerten komplett automatisiert (Abdou u. Pointon, 2011).

Das vorangegangene Beispiel beinhaltet typische Elemente einer Klassifikation: Instanzen, Features und Klassen. Eine Instanz ist im Beispiel ein Kunde. Jede Instanz wird durch dieselben Attribute – aber womöglich mit unterschiedlichen Ausprägungen – beschrieben. Diese Attribute werden als Features, Φ , bezeichnet. Die Ausprägungen der Features für eine gegebene Instanz dienen als Input-Vektor X für die Klassifizierung, wie in Gleichung 2.1 dargestellt. Aufgrund der grundlegenden Bedeutung der Features stellen deren Berechnung (Feature Extraction) und Auswahl (Feature Selection) zentrale Schritte im Machine Learning dar. Die möglichen kategorischen Werte der Ergebnisvariable y , genannt Klassen, könnten im Beispieldfall "Kunde mit hohem Risiko" und "Kunde mit geringem Risiko" sein. Das Ergebnis einer Klassifikation ist die Klasse der betrachteten Instanz, im Beispiel die Kreditwürdigkeit eines Kunden.

Ein simpler Klassifikator könnte unter anderem nach folgender Regel handeln:

$$IF(\text{Salary} > 100.000 \wedge \text{Credit Amount} < 200.000), THEN \text{ Class} = \text{"Low Risk"}. \quad (2.7)$$

Basierend auf dem Gehalt des Kunden und der Kreditsumme, schätzt das Modell das Risiko der Kreditvergabe ein. In der Realität reichen diese zwei Kriterien allerdings häufig nicht aus, um eine fundierte Entscheidung zu treffen. Bankangestellte prüfen ebenfalls die Kredithistorie, die Lebenssituation, den Arbeitsvertrag, den persönlichen Eindruck sowie weitere – möglicherweise auch die Persönlichkeit betreffende – Faktoren. Sofern diese Faktoren als Features in den Daten vorhanden sind, eignet sich der Klassifikator diese in der Trainingsphase an und ergänzt sie in das Entscheidungsmodell. Bis zu einem gewissen Punkt erhöht das Hinzufügen von Features und Regeln den Erfolg des Modells, wie in Abschnitt 2.7 dargestellt, bevor die zunehmende Komplexität die Generalisierungsfähigkeit vermindert.

Diese Bachelorarbeit beschränkt sich auf binäre Klassifikatoren, also Klassifikatoren mit genau zwei Klassen. Konkret werden Decision Tree und Random Forest Klassifikatoren untersucht. Es folgen nun die theoretischen Grundlagen der Erstellung, Anwendung und Erfolgsmessung dieser Modelle.

2.3 Decision Tree Klassifikator

2.3.1 Generelle Funktionsweise und Eigenschaften

Die Induktion von Decision Trees (Quinlan, 1986) gehört zu den simpelsten aber erfolgreichsten Machine Learning Algorithmen (Russell u. Norvig, 2009). Gupta u. a. (2017) nennen unter anderem die medizinische Diagnostik, intelligente Fahrzeuge, das Credit Scoring (siehe Abschnitt 2.2) und die industrielle Qualitätskontrolle als Anwendungsbereiche. Außerdem bilden Decision Trees die Grundlage für das später behandelte Random Forest Ensemble. Ein Decision Tree funktioniert nach dem Teile-und-Herrsche Prinzip. Die Trainingsstichprobe $\chi_{training}$ wird nach einem festgelegten Kriterium in Teilmengen geteilt. Die entstandenen Teilmengen wiederum werden nach der gleichen Prozedur aufgeteilt. Diese Rekursion findet so lange statt, bis ein definiertes Endkriterium erfüllt ist und der Decision Tree zur Klassifikation bereit ist. Nachfolgend wird zuerst der Aufbau eines Decision Trees erklärt, anschließend dessen Erstellung und Optimierung.

Anknüpfend an das Beispiel der Kreditvergabe zeigt Abbildung 2.1 einen möglichen Decision Tree für diesen Anwendungsfall.

Ein Decision Tree besteht aus der Wurzel, aus internen Knoten, aus Ästen und aus externen Knoten, genannt Blätter. Die Wurzel ist der einzige interne Knoten, der keinen Vorgänger hat. Ein interner Knoten ist ein Knoten, der Nachfolger hat. Interne Knoten testen jeweils ein bestimmtes Feature Φ der betrachteten Instanz X auf dessen Wert. Äste sind die Verbindungen zwischen Knoten. Jeder Ast bildet eine Wertemenge ab, die im vorgelagerten internen Knoten festgelegt wurde. Ein Blatt ist ein Knoten ohne Nachfolger. Blätter repräsentieren, im Fall der Klassifikation, die Klassen (Quinlan, 1986). In dieser Arbeit werden binäre Decision Trees betrachtet, die genau zwei Klassen kennen. Auch finden ausschließlich univariate Bäume Anwendung, das heißt, interne Knoten testen jeweils nur auf ein Feature, und nicht auf mehrere.

Die Klassifikation einer Instanz beginnt in der Wurzel des Decision Trees. Dort wird die Instanz auf ein Feature getestet und, je nach Wert dieses Features, entweder an den linken oder an den rechten Nachfolger der Wurzel weitergeleitet. Diese Prüfung mit Weiterleitung findet solange statt, bis die Instanz an einem Blatt angekommen ist. Dort wird der Instanz die Klasse des Blattes zugewiesen, wodurch diese Instanz klassifiziert ist (Russell u. Norvig, 2009). Der Graph aller Knoten, die die Instanz durchwandert hat, nennt sich Pfad.

Eine vorteilhafte Eigenschaft des Decision Trees ist die Best Case Laufzeit-Komplexität von $\mathcal{O}(m \times n \times \log_2(n))$ zur Erstellung, die erreicht wird, wenn die Baumstruktur balanciert ist (Gron, 2017). Die Variable m steht für die Anzahl an Features, n steht für die Anzahl der Instanzen, mit denen der Decision Tree erstellt wurde, also $|\chi_{training}|$. Im Worst Case liegt die Komplexität bei $\mathcal{O}(n^2)$. Das ist der Fall, wenn für jedes ϕ_{opt} an jedem internen Knoten die Instanzen $X \in \chi_{training}$ so aufgeteilt werden, dass eine einzige Instanz als Blatt deklariert

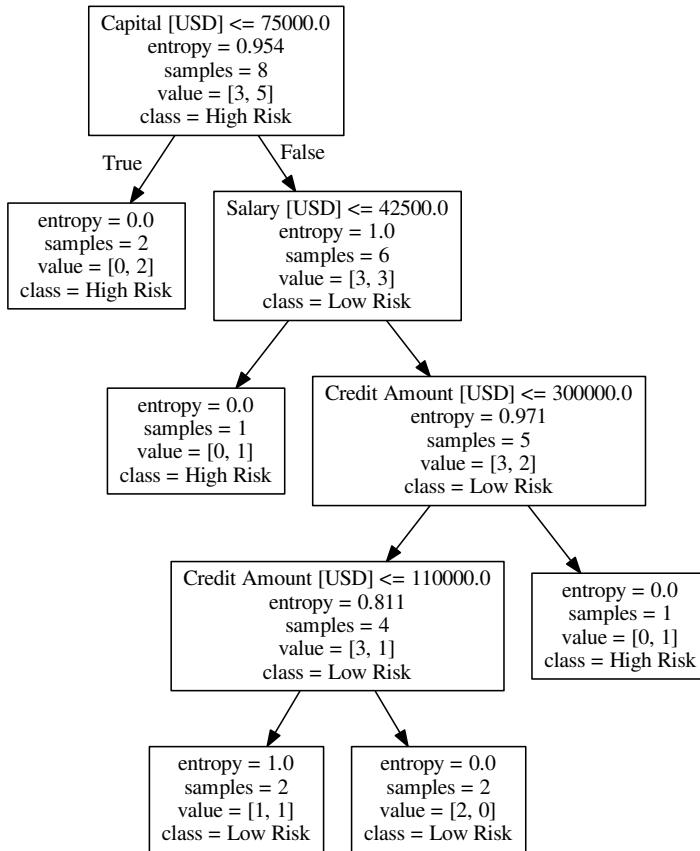


Abbildung 2.1 Beispielhafter Decision Tree für den Anwendungsfall der Kreditvergabe

wird und die restlichen $n - 1$ Instanzen in einen weiteren internen Knoten einfließen. Dort wiederholt sich der Vorgang rekursiv solange, bis alle Instanzen einem Blatt zugewiesen wurden und der Decision Tree fertig erstellt ist.

Ein weiterer Vorteil von Decision Trees ist, dass sie eine Menge von geordneten simplen Regeln darstellen und somit für den Menschen leicht verständlich sind, wie die Regel aus dem Abschnitt 2.2 zeigt. So lässt sich zum Beispiel die Regel

$$\begin{aligned} & IF(Capital > 75.000 \wedge Salary > 42.500 \wedge CreditAmount > 300.000), \\ & THEN Class = "HighRisk" \end{aligned} \quad (2.8)$$

aus der Abbildung 2.1 herleiten. Diese Verständlichkeit macht den Decision Tree zu einem sogenannten White Box Modell, dessen Entscheidungen für den Menschen nachvollziehbar sind. Black Box Modelle hingegen, wie der Random Forest oder neuronale Netze, treffen schwer- oder nicht-rekonstruierbare Entscheidungen (Gron, 2017). Wegen dieser Vorteile ist der Decision Tree Klassifikator sehr beliebt und wird in manchen Fällen den exakteren aber komplexeren Methoden vorgezogen (Alpaydin, 2008).

Im Gegensatz zu einigen statistischen Modellen wie der linearen oder exponentiellen Regression, gibt es im Funktionenraum F von Decision Trees stets eine Funktion $f(X|\Theta)$, die die Klassen aller Trainingsinstanzen korrekt abbildet. Ein Decision Tree kann prinzipiell beliebig viele Regeln definieren und somit jede Instanz aus $\chi_{training}$ korrekt abbilden. Diese Eigenschaft bringt allerdings einen Nachteil mit sich. Decision Trees sind für Overfitting (siehe Abschnitt 2.7) anfällig. Denn falls der Decision Tree, ohne Kontrolle der Baumstruktur, für jede Instanz einen eigenen Blattknoten anlegt, erzielt er zwar auf $\chi_{training}$ ein optimales Ergebnis, ist jedoch nicht zur Generalisierung und somit nicht zum Lernen fähig. Entsprechende Gegenmaßnahmen, um die Baumstruktur zu kontrollieren, finden sich im Unterabschnitt 2.3.2.

2.3.2 Erstellung eines Decision Trees

Die Induktion anhand des ID3-Algorithmus führt zu einem von mehreren äquivalenten Decision Trees. Es gibt mehr als einen Decision Tree, der die Regeln einer gegebenen Stichprobe $\chi_{training}$ kodiert (Quinlan, 1986). Nach Ockhams Rasiermesser ist es erstrebenswert, den Baum mit der geringsten Komplexität den anderen vorzuziehen (siehe Unterabschnitt 2.3.4). Diesen zu suchen ist jedoch ein NP-vollständiges Problem (Quinlan, 1986). Stattdessen bietet sich eine lokale Suche über Heuristiken an. Die nachfolgenden Lernalgorithmen sind vom Greedy-Typ: Von der Wurzel aus wählt der Algorithmus in jedem Schritt die in der aktuellen Position beste Aufteilung der Instanzen, um den Decision Tree iterativ zu erstellen (Alpaydin, 2008). Es folgt eine Beschreibung dieser Aufteilung sowie, anschließend, der Güte einer gegebenen Aufteilung.

Beginnend mit allen Trainingsinstanzen $\chi_{training}$ an der Wurzel, entsteht der Decision Tree durch rekursive Aufteilungen derselben an seinen internen Knoten. Eine Aufteilung bezeichnet hier das Bilden von Teilmengen der betrachteten Trainingsinstanzen an einem internen Knoten K_0 , um pro Teilmenge einen Ast sowie einen weiteren Knoten K_1 zu generieren. Eine Aufteilung erfolgt stets anhand des optimalen Features ϕ_{opt} . Es sind zwei Fälle zu unterscheiden: Entweder ist das Feature diskret oder es ist numerisch. Ist das Feature diskret, wird für jede Ausprägung dieses Features ein Ast generiert. Ist das Feature numerisch, wird es diskretisiert. Dazu werden ein oder mehrere Schwellenwerte festgelegt, die entsprechende Teilmengen definieren. Weist ein Feature ϕ_i zum Beispiel die Werte in $[v_{start}, v_{end}]$ auf, könnte der Algorithmus die Schwellenwerte $\{s_1, s_2\}$ so wählen, dass gilt

$$v_{start} < s_1 < s_2 < v_{end}. \quad (2.9)$$

Die entstehenden Teilmengen wären dann

$$\begin{aligned} V_1 &= \{v | v < s_1\}, \\ V_2 &= \{v | v \geq s_1 \wedge v < s_2\} \text{ und} \\ V_3 &= \{v | v \geq s_2\} \end{aligned} \quad (2.10)$$

und würden zu drei entsprechenden Ästen führen. Der Spezialfall von genau einem Schwellenwert, wodurch sich zwei Teilmengen ergeben, wird als binäre Aufteilung bezeichnet (Alpaydin, 2008). Binäre Aufteilungen führen graphisch gesehen zu Hyperrechtecken, die die Instanzen voneinander abgrenzen.

Die Anzahl der möglichen Aufteilungen ist gleich der Anzahl der vorhandenen Features, m . Die möglichen Aufteilungen unterscheiden sich hinsichtlich ihres Einflusses auf den finalen Decision Tree. Deshalb wird die Güte der möglichen Aufteilungen verglichen, um die sinnvollste von ihnen zu identifizieren.

Als Kriterium für die Güte der Aufteilung verwendet der ID3-Algorithmus die aus ihr resultierende Änderung in Entropie, bezeichnet als Information Gain (IG) (Quinlan, 1986). Die Entropie H liegt per Definition zwischen Null und Eins und trifft eine Aussage über die Homogenität einer gegebenen Menge an n Instanzen. In der Informationstheorie bezeichnet die Entropie "die minimale Anzahl an Bits, die nötig sind, um die Klassifikationsgenauigkeit einer Instanz zu codieren" (Alpaydin, 2008). An jedem Knoten K wird die Entropie berechnet mit (Shannon, 1948)

$$H(K) = - \sum_w p_i \log_2(p_i), \quad (2.11)$$

wobei w für die möglichen Ausprägungen von Φ_{opt} steht. Die Variable p_i bezeichnet die relative Häufigkeit der Klasse y_i innerhalb jener Instanzen, deren Ausprägung von Φ_{opt} gleich w_i ist. Besitzen alle Instanzen mit der Ausprägung w_i dieselbe Klasse, so ist die Entropie Null (da $\log_2(1) = 0$) und der Knoten bekommt keine Nachfolger, sondern wird zum Blatt. Besitzt jedoch mindestens eine Instanz eine andere Klasse, so ist die Entropie größer als Null. Dann sucht ID3 nach jener Aufteilung, welche den größten IG mit sich bringt. Angenommen die Aufteilung erfolgt anhand von Feature ϕ_j . Dann ergibt sich der IG zwischen einem Knoten K_d der Tiefe d und seinen potenziellen Nachfolgern $K_d|\phi_j$ der Tiefe $d + 1$ aus

$$IG(K_d, K_d|\phi_j) = H(K_d) - H(K_d|\phi_j). \quad (2.12)$$

Weiterhin angenommen, dass $\phi_j k$ Ausprägungen vorweist, dann folgt die Entropie der potenziellen Nachfolger der Tiefe $d + 1$ aus

$$H(K_d|\phi_j) = \sum_k P(\phi_j = v_k) H(K_d|\phi_j = v_k), \quad (2.13)$$

wobei $P(\phi_j = v_k)$ die Wahrscheinlichkeit bezeichnet, dass ϕ_j den Wert v_k annimmt, und $H(K_d|\phi_j = v_k)$ die erwartete Entropie an jenem Knoten in Tiefe $d + 1$ bezeichnet, an welchen eben diese Instanzen mit der Ausprägung v_k gelangen. Diese erwartete Entropie ergibt sich wiederum aus der Gleichung 2.11.

Da der ID3-Algorithmus den IG maximiert, minimiert er die erwartete Anzahl an Schritten, die von der Wurzel bis zum Blatt nötig sind, also die Tiefe des Baumes. Eine minimale Tiefe wiederum bedeutet eine geringere Komplexität und ist nach Ockhams Rasiermesser den Alternativen vorzuziehen. Außerdem beschleunigt eine geringe Tiefe die Klassifikation, da die Instanz weniger Knoten und somit weniger Tests durchlaufen muss, bevor sie einen Blattknoten erreicht und dessen Klasse zugewiesen bekommt. Da der beschriebene Greedy-Algorithmus nur lokal sucht, ist das Ergebnis allerdings nicht zwangsläufig das globale Optimum. Die Instanzen an allen internen Knoten in jeder Tiefe d werden nach dem beschriebenen Procedere aufgeteilt und den entstandenen Nachfolger-Knoten in Tiefe $d + 1$ zugewiesen. Dort ruft sich der Algorithmus rekursiv auf. Sobald alle Instanzen an einem Knoten dieselbe Klasse aufweisen, wird dieser Knoten ein Blatt und die Instanzen werden nicht weiter aufgeteilt. Dem Blatt wird die Mehrheitsklasse seiner Instanzen zugewiesen. Somit ist ein neuer Pfad, von der Wurzel bis zum Blatt, im Decision Tree entstanden. Pseudocode

zur Erstellung eines Decision Trees ist in Algorithmus 1 zu finden. Die Laufzeitkomplexität zur Erstellung eines balancierten Modells mit n Instanzen liegt bei $\mathcal{O}(m \times n \times \log_2(n))$, da an jedem Knoten alle m Features verglichen werden (Gron, 2017). Die Klassifikation selbst nimmt dann $\mathcal{O}(\log_2(n))$ in Anspruch, da die Instanz pro internem Knoten nur auf ein einziges Feature getestet wird.

Tabelle 2.1 (Hyper-)Parameter eines Decision Trees und deren Bedeutung

Name	Typ	Beschreibung	Zweck (Beispiele)
Maximale Anzahl an betrachteten Features	H	Beschränkung der Features, die pro Knoten verglichen werden	Beschleunigung des Trainings
Maximale Tiefe	H	Begrenzung der Tiefe des Decision Trees	Reduktion von Overfitting
Aufteilungskriterium	H	Kriterium zur Auswahl von Φ_{opt} für eine Aufteilung an Knoten K, z.B. IG oder Gini Index	Messung der Reinheit von Instanzenmengen
Minimale Anzahl an Instanzen für Aufteilung	H	Schwellenwert bezüglich der Anzahl an Instanzen für eine weitere Aufteilung der Instanzen	Reduktion von Overfitting
Minimale Anzahl an Instanzen für Blatt	H	Schwellenwert bezüglich der Anzahl an Instanzen für die Erstellung eines neuen Blattes	Reduktion von Overfitting
Maximale Anzahl an Blättern	H	Begrenzung der Menge an Blättern im Decision Tree	Reduktion von Overfitting
Minimale Unreinheitsreduktion für Aufteilung	H	Schwellenwert bezüglich des IG für eine weitere Aufteilung der Instanzen	Reduktion von Overfitting
Gewichtung der Instanzen in $\chi_{training}$	H	Gewichtung der Beispieldaten für das Training	Hervorhebung repräsentativer Instanzen bzw. Unterdrückung von Ausreißern
Tiefe	P	Tiefe des Decision Trees, Anzahl der Stufen von Wurzel bis zum entferntesten Blatt	Beurteilung der finalen Struktur
Anzahl an Blättern	P	Anzahl der Blätter im finalen Decision Tree	Beurteilung der finalen Struktur
Anzahl an internen Knoten	P	Anzahl an Knoten mit Nachfolgern (inklusive Wurzel, exklusive Blätter)	Beurteilung der finalen Struktur
Signifikanzen der Features	P	Bedeutung jedes Features Φ_i bei der Klassifizierung, z.B. berechnet anhand des IG durch Φ_i	Identifizierung der einflussreichsten Features
Anzahl der Features	P	Anzahl der Features, auf die die internen Knoten des Decision Trees testen	Beurteilung der finalen Struktur
Anzahl der Klassen	P	Anzahl der Klassen, die die Blätter des Decision Trees abbilden	Beurteilung der finalen Struktur

Russell u. Norvig (2009) weisen darauf hin, dass Features mit einem starken Verzweigungsfaktor nach dem ID3-Verfahren bevorzugt werden. Ein Beispiel dafür ist ein Zeitstempel, der für jede Instanz aus $\chi_{training}$ einzigartig ist. Dann würde eine Aufteilung anhand dieses Features stets zu einer erwarteten Entropie von Null führen, und würde somit stets den größtmöglichen IG erzielen. Der ID3-Algorithmus bevorzugt ein solches Feature. Das kann

zu starkem Overfitting führen, da das Modell nicht zur Generalisierung befähigt wird, also nicht lernt. Solchen stark-verzweigenden Features kann mit einer Bestrafung basierend auf dem Verzweigungsfaktor, zum Beispiel mithilfe des Gain Ratios, entgegengewirkt werden (Russell u. Norvig, 2009). Ein Nachfolger des ID3-Algorithmus namens C4.5 berücksichtigt den Gain Ratio bei der Erstellung des Decision Trees (Gupta u. a., 2017).

2.3.3 Grundlegende (Hyper-)Parameter

Ein Machine Learning Algorithmus optimiert die Parameter Θ , um die Fehlerrate des Klassifikators zu minimieren (siehe Abschnitt 2.1). Die optimalen Werte, Θ_{opt} , resultieren also aus dem Trainingsprozess und sind erst im Nachhinein bekannt. Neben diesen Parametern gibt es Hyperparameter, die vom Anwender zu Beginn zu setzen sind. Einige dieser Hyperparameter beeinflussen die finalen Parameter Θ_{opt} , zum Beispiel indem sie deren Wertebereiche einschränken. Die Tabelle 2.1 orientiert sich an der Implementierung von Scikit-learn (Pedregosa u. a., 2011) und gibt einen Überblick über grundlegende Parameter (Typ in der Tabelle ist "P") sowie Hyperparameter (Typ in der Tabelle ist "H") von Decision Trees und deren Bedeutung, einschließlich möglicher Zwecke, zu denen diese Hyperparametern verändert werden.

2.3.4 Optimierung von Decision Trees

Nach der Erstellung ist ein Decision Tree oft zu komplex und neigt zu Overfitting (Gron, 2017). Eine Methode zur Reduzierung von Overfitting ist das Pruning, also das Kürzen des Baumes. Es wird zwischen Prepruning und Postpruning unterschieden. Prepruning findet noch während der Erstellung des Decision Trees statt. Ein Ansatz für Prepruning ist es, eine Bedingung für Aufteilungen an jedem Knoten zu definieren. Dadurch soll verhindert werden, dass Entscheidungen, die auf zu wenigen Instanzen basieren, die Varianz des Klassifikators erhöhen und somit die Generalisierung verschlechtern (Alpaydin, 2008). So könnte man zum Beispiel festlegen, dass eine weitere Aufteilung nur dann stattfindet, wenn mindestens fünf Prozent der Trainingsinstanzen aus $\chi_{training}$ zu diesem Knoten gelangt sind. Andernfalls wird dieser Knoten zu einem Blatt, betitelt mit der Mehrheitsklasse.

Postpruning hingegen verändert den Baum, nachdem er komplett erstellt wurde. Die Decision Tree Pruning-Methode versucht jene Teilbäume zu identifizieren, welche für das Overfitting verantwortlich sind. Dazu wird vor dem Erstellen des Decision Trees eine Pruning-Menge $\chi_{pruning} \subsetneq \chi$ festgelegt, welche nicht in das Training einfließt. Das Training findet stattdessen mit den Instanzen aus $\chi \setminus \chi_{pruning}$ statt. Anschließend erfolgen pro Teilbaum zwei Messungen von Fehlerraten auf $\chi_{pruning}$. In der ersten Version klassifiziert der Decision Tree in seiner finalen Form die Instanzen aus $\chi_{pruning}$, ohne Veränderung des betrachteten Teilbaums. In der zweiten Version wird der jeweils betrachtete Teilbaum mit seiner Mehrheitsklasse ersetzt, also als Blatt simuliert. Ist der Fehler in der zweiten Version nicht signifikant schlechter als jener in der ersten Version, so wird der betrachtete Teilbaum dauerhaft in ein Blatt umgewandelt. Damit verringert sich die Komplexität und somit das Overfitting des Decision Trees (Russell u. Norvig, 2009). In der Praxis hat sich Postpruning als zielführender als Prepruning herausgestellt (Alpaydin, 2008).

2.4 Ensemble Methoden

2.4.1 Die Ensemble-Idee

Nach dem No-free-lunch Theorem ist kein Lernalgorithmus besser als ein anderer, sofern die durchschnittliche Fehlerrate über alle möglichen diskreten Funktionen betrachtet wird (Whitley u. Watson, 2005). Es gibt also keinen einzigen Lernalgorithmus, der in jedem Anwendungsfall das genaueste Modell erzeugt (Alpaydin, 2008). Ist der beste Klassifikator in einer gegebenen Situation gesucht, so wird dieser beispielsweise mit einem Validierungsdatenset χ_{val} ermittelt. Eine Alternative, die die Auswahl eines einzigen besten Klassifikators umgeht, ist das Ensemble Learning. Die Idee dabei ist es, die – nicht zu 100% ausmerzbaren – Fehler an einer Instanz X_i von einzelnen Klassifikatoren durch andere Klassifikatoren, die X_i korrekt abbilden, überzukompensieren (Russell u. Norvig, 2009). Die einzelnen Basisklassifikatoren verhalten sich dann komplementär zueinander. Das Ziel ist es, dadurch die Fehlerrate insgesamt im Ensemble zu reduzieren.

Ein Ensemble ist ein Klassifikator, der aus mindestens zwei anderen, individuellen Klassifikatoren besteht. Die individuellen Klassifikatoren werden als Basisklassifikatoren bezeichnet. Als Basisklassifikatoren eignen sich Modelle, die eine niedrigere Fehlerrate als zufälliges Raten erreichen. Ist dieses Kriterium erfüllt, so führt zum Beispiel das nachfolgend behandelte Boosting zu Ensembles mit "beliebig hoher Treffergenauigkeit" (Freund u. Schapire, 1997). Um eine möglichst große Fehlerreduktion im Vergleich zu den einzelnen Basisklassifikatoren zu erreichen, ist es nötig, diese Basisklassifikatoren maximal unterschiedlich in ihrer Entscheidungsfindung zu gestalten, sie also zu dekorrelieren. Zur Messung der Dekorrelation von Decision Trees schlägt Tin Kam Ho (1998) zum Beispiel die Metrik Tree Agreement vor. Für die Dekorrelation gibt es verschiedene Ansätze. So können zum Beispiel verschiedene Lernalgorithmen genutzt werden, dieselben Lernalgorithmen mit unterschiedlichen Hyperparametern belegt werden, oder die Basisklassifikatoren mit unterschiedlichen Repräsentationen der Input-Daten trainiert werden (Alpaydin, 2008). Eine weitere Möglichkeit zur Dekorrelation der Basisklassifikatoren ist das Variieren der Trainingsdaten. Dazu gehören die Ansätze des Bagging und Boosting. Diese behandelt der nachfolgende Unterabschnitt 2.4.2.

2.4.2 Bagging und Boosting

Zwei verbreitete Methoden zur Dekorrelation von Basisklassifikatoren in Ensembles sind das Bagging und das Boosting. Beide Methoden können sowohl auf Klassifikatoren als auch auf Regressionsmodelle angewendet werden (Alpaydin, 2008). Bagging steht für Bootstrap Aggregation. Nach dieser Methode werden für jeden Basisklassifikator mit dem Bootstrap-Algorithmus zufällige Instanzen aus χ ausgewählt. Das heißt, der jeweilige Basisklassifikator β_i wird mit der Stichprobe $\chi_i \subset \chi$ trainiert, wobei $|\chi_i| = |\chi|$. Diese Auswahl erfolgt mit Wiederholung; eine Instanz aus χ kann also mehrmals in χ_i vorkommen oder auch gar nicht. Die Wahrscheinlichkeit, dass eine Instanz X nicht in die Bootstrap-Stichprobe χ_i kommt ist (Alpaydin, 2008)

$$P(X \notin \chi_i) = \left(1 - \frac{1}{|\chi|}\right)^{|\chi|} \approx e^{-1} = 36,8\%. \quad (2.14)$$

Das heißt im Umkehrschluss, dass die Bootstrap-Stichprobe χ_i ungefähr $100\% - 36,8\% = 63,2\%$ der Instanzen aus χ enthält. Im Gegensatz zum Boosting entstehen die Stichproben χ_i beim Bagging unabhängig voneinander.

Boosting bedeutet wörtlich Verstärken. Dabei sind die Stichproben nicht unabhängig voneinander, sondern werden der Reihe nach gebildet. Die Idee von Boosting ist es, die Stichprobe χ_{i+1} dahingehend anzupassen, dass sie jene Instanzen aus χ_i bevorzugt, welche von dem darauf trainierten Basisklassifikator nicht erfolgreich gelernt wurden. Es werden also jene Instanzen bevorzugt, für welche das Modell die falsche Klasse bestimmt hat. Der Basisklassifikator β_{i+1} soll also von den Fehlern seines Vorgängers β_i lernen. Somit sollen die Basisklassifikatoren ihre Schwächen untereinander ausgleichen. Um den Bedarf nach einer sehr großen Datenmenge zum Boosting zu beseitigen, entwarfen Freund u. Schapire (1997) die Variante AdaBoost, oder ausgeschrieben Adaptive Boosting. Außerdem ist AdaBoost, im Gegensatz zu seinen Vorgängern, dazu in der Lage, beliebig viele Basisklassifikatoren zu kombinieren (Alpaydin, 2008).

2.4.3 Entscheidungsfindung im Ensemble

Die Entscheidung, welche Klasse einer neuen Instanz X zugewiesen wird, trifft ein Ensemble auf Basis der Entscheidungen seiner Basisklassifikatoren. Ein Ensemble E , bestehend aus den n Basisklassifikatoren $\{\beta_1, \dots, \beta_n\}$ unter Verwendung von Kombinationsmethode Ψ , kann als Vorhersagefunktion für Instanz X als

$$E(X|\{\beta_1, \dots, \beta_n\}, \Psi) = \hat{y} \quad (2.15)$$

formalisiert werden. Für Ψ lassen sich einstufige und mehrstufige Kombinationsmethoden unterscheiden (Alpaydin, 2008). Diese Arbeit beschränkt sich auf einstufige Methoden. Eine solche einstufige Methode ist die nachfolgend beschriebene Voting-Methode.

Wenn $\{\hat{y}_1, \dots, \hat{y}_n\}$ die Ergebnisse von $\{\beta_1, \dots, \beta_n\}$ sind, dann ergibt sich nach der Voting-Methode Ψ_{voting} die Entscheidung des Ensembles, \hat{y}_E , allgemein – im Sinne von sowohl für Klassifikation als auch für Regression gültig – mit

$$\hat{y}_E = \Psi_{voting}(\hat{y}_1, \dots, \hat{y}_n). \quad (2.16)$$

Handelt es sich bei $\{\hat{y}_1, \dots, \hat{y}_n\}$ um die vorhergesagten Klassen, so nennt man dies Hard Voting. Handelt es sich dabei jedoch um die Wahrscheinlichkeiten für die positive Klasse, so spricht man von Soft Voting (Gron, 2017). Im Spezialfall, dass das Ensemble als Klassifikator verwendet wird, entspricht das finale Ergebnis der nächstgelegenen – gemessen an der Distanz zu \hat{y}_E – Klasse. Das heißt in einer binären Klassifikation mit den Klassen $y \in \{0, 1\}$ berechnet sich das Ergebnis \hat{y}_{Class} mit

$$\hat{y}_{Class} = \begin{cases} 1 & (\text{positiv}), \quad \text{wenn } 0,5 \leq \hat{y}_E \leq 1 \\ 0 & (\text{negativ}), \quad \text{wenn } 0 \leq \hat{y}_E < 0,5. \end{cases} \quad (2.17)$$

Die Annahme bezüglich Ψ_{voting} ist dabei, dass alle n Basisklassifikatoren $\{\beta_1, \dots, \beta_n\}$ gleich gewichtet sind, und zwar mit einem Gewicht von $\frac{1}{n}$. In der Literatur finden sich Vorschläge,

wie man die Basisklassifikatoren gewichten kann. Ein Beispiel dafür ist es, die Basisklassifikatoren auf einem Validierungsdatenset χ_{val} zu bewerten, um sie dann im Ensemble anhand ihrer Treffgenauigkeit auf χ_{val} zu gewichten (Alpaydin, 2008).

An dieser Stelle sei, alternativ zur Voting-Methode, die geschachtelte Generalisierung von Wolpert (1992) erwähnt. Die Idee der geschachtelten Generalisierung ist es, die Kombination von $\hat{y}_1, \dots, \hat{y}_n$ nicht durch eine (gewichtete) Summe zu vollziehen, sondern selbst wiederum durch eine lernende Funktion Ψ_{sg} zu entwerfen. Das finale Ergebnis eines Ensembles, das eine solche Kombinationsfunktion verwendet, lautet dann

$$\hat{y}_E = \Psi_{sg}(\hat{y}_1, \dots, \hat{y}_n | \Theta). \quad (2.18)$$

Ein Lernalgorithmus sucht die optimalen Parameter Θ_{opt} , sodass die Fehlerfunktion des Ensembles minimiert wird, wie in Abschnitt 2.1 beschrieben. Die Kombinationsfunktion Ψ_{sg} soll also die Schwächen der einzelnen Basisklassifikatoren lernen, um diese bei der finalen Kombination zu berücksichtigen (Alpaydin, 2008). Da nun sowohl die Base Classifier, die $\{\hat{y}_1, \dots, \hat{y}_n\}$ erzeugen, als auch die Kombinationsfunktion Ψ_{sg} selbst anhand von Daten lernen, findet eine geschachtelte Generalisierung statt.

2.5 Random Forest Klassifikator

2.5.1 Generelle Funktionsweise und Eigenschaften

Ein Random Forest ist ein Ensemble, das Decision Trees als Basisklassifikatoren nutzt. Trotz ihrer vergleichsweise simplen Struktur gehören Random Forests zu den mächtigsten Machine Learning Algorithmen (Gron, 2017). Durch das Einführen von Zufälligkeit ist ein Random Forest robuster als ein einzelner Decision Tree. Die Zufälligkeit entsteht durch Bagging und durch zufällige Auswahl jener Attribute, die für einen Split überhaupt erst betrachtet werden. Diese Zufälligkeit hilft dabei, unkorrelierte Decision Trees zu generieren, die ihre Schwächen gegenseitig ausgleichen und somit komplementär zueinander sind.

Die Klassifikation durch einen Random Forest erfolgt nach der Voting-Methode (siehe Unterabschnitt 2.4.3). Die Decision Trees aus dem Random Forest klassifizieren die Instanz nach dem in Unterabschnitt 2.3.1 dargelegten Procedere. Die Ergebnisse fließen als gleichgewichtete Stimmen in den Random Forest ein. Die am häufigsten vorkommende Klasse ist das Ergebnis der Klassifizierung.

Der Funktionenraum F eines Random Forests enthält – wie auch der von Decision Trees – stets eine Funktion f , die die Klassen aller Trainingsinstanzen korrekt abbildet. Der entscheidende Unterschied zu Decision Trees jedoch ist die Lösung des Overfitting-Problems. Mithilfe des Gesetzes der großen Zahlen hat Breiman (2001) bewiesen, dass bei Random Forests kein Overfitting auftritt. Der Generalisierungsfehler eines Random Forests hängt von der Güte der einzelnen Decision Trees sowie von der Korrelation dergleichen ab (Breiman, 2001).

Eine vorteilhafte Eigenschaft ergibt sich aus der Tatsache, dass der Random Forest Bagging verwendet. Wie in Kapitel 2.4.2 gezeigt, enthalten die Stichproben, auf denen die Basisklassifikatoren trainiert werden, durchschnittlich 63,2% der Instanzen aus χ . Die restlichen 36,8%

der Instanzen aus χ werden als out-of-bag (OOB) bezeichnet. Diese OOB Instanzen können zur Erfolgsmessung verwendet werden, noch bevor ein Validierungs- oder Testdatenset zum Einsatz kommt. Die gemessene Fehlerrate nennt sich in dem Fall OOB Error (Gron, 2017).

Algorithm 1 Random Forest Pseudocode

Require: Trainingset χ , Features Φ , Anzahl der Decision Trees n

```

1: function RandomForest ( $\chi_{random\_forest}, \Phi_{random\_forest}, n$ )
2:  $T \leftarrow \emptyset$ 
3: for  $i \in \{1, \dots, n\}$  do
4:    $\chi_i \leftarrow$  Bootstrapping-Set von  $\chi_{random\_forest}$ 
5:    $\Phi_i \leftarrow$  Randomisierte, echte Teilmenge von  $\Phi_{random\_forest}$ 
6:    $t_i \leftarrow$  DecisionTree( $\chi_i, \Phi_i$ )
7:    $T \cup t_i$ 
8: end for
9: return  $T$ 
10: end function

11: function DecisionTree( $\chi_{decision\_tree}, \Phi_{decision\_tree}$ )
12:  $K \leftarrow$  Wurzel-Knoten  $k_1$ 
13: while  $K$  enthält mindestens einen internen Knoten  $k_{intern\_neu}$  ohne Nachfolger do
14:    $\Phi_{opt} \leftarrow$  Feature mit höchstem IG an Knoten  $k_{intern}$ 
15:   Teile Instanzen an  $k_{intern\_neu}$  entsprechend der Werte von  $\Phi_{opt}$  auf neue Knoten  $K_{neu}$  auf
16:   Weise  $k_{intern\_neu}$  alle Knoten aus  $K_{neu}$  als Nachfolger zu
17:   for  $K_{neu\_i} \in K_{neu}$  do
18:     if Instanzen an Knoten  $K_{neu\_i}$  haben alle dieselbe Klasse then
19:       Erkläre  $K_{neu\_i}$  als Blatt mit Mehrheitsklasse
20:     else
21:       Erkläre  $K_{neu\_i}$  als internen Knoten ohne Nachfolger
22:     end if
23:   end for
24:    $K \cup K_{neu}$ 
25: end while
26: return  $K$ 
27: end function

```

2.5.2 Erstellung eines Random Forests

Ein Random Forest entsteht durch das Trainieren von n Decision Trees auf den Trainingssets $\{\chi_1, \chi_n\}$, welche mittels Bagging aus χ generiert werden.

Eine Besonderheit dabei ist, dass nicht alle Features für die Aufteilung an Knoten K betrachtet werden. Stattdessen erwägt der Random Forest Algorithmus eine zufällige, echte Untermenge der Features. Von den i Attributen der Daten wird eine Untermenge von j Attributten zufällig ausgewählt. Für diese j Attribute werden im nächsten Schritt jeweils die IG-Werte berechnet, die aus ihren Aufteilungen (siehe Unterabschnitt 2.3.2) resultieren würden. Das Attribut mit dem höchsten IG wird für den betrachteten Knoten als Auftei-

lungskriterium gewählt. Diese Zufälligkeit dekorreliert die Basisklassifikatoren, denn diese treffen ihre Entscheidungen basierend auf verschiedenen Features, und ermöglicht dadurch ein komplementäres Ensemble. Algorithmus 1 veranschaulicht die Erstellung eines Random Forests mittels Pseudocode.

2.5.3 Grundlegende (Hyper-)Parameter

Zusätzlich zu den (Hyper-)Parametern seiner Basisklassifikatoren (siehe Unterabschnitt 2.3.3) besitzt der Random Forest als Ensemble noch weitere Einstellungen. Die Tabelle 2.2 orientiert sich an der Implementierung von Scikit-learn (Pedregosa u. a., 2011) und gibt einen Überblick über relevante (Hyper-)Parameter.

Tabelle 2.2 (Hyper-)Parameter eines Random Forests und deren Bedeutung

Name	Typ	Beschreibung	Zweck (Beispiele)
Anzahl der Basisklassifikatoren	H	Anzahl der Decision Trees innerhalb des Random Forests	Steigerung der Treffergenauigkeit durch mehr komplementäre Basisklassifikatoren
Bootstrap-Sampling	H	Bestimmung, ob Basisklassifikatoren auf den gesamten Daten oder auf Bootstrap-Stichproben trainiert werden	Steigerung der Zufälligkeit bei Training der Basisklassifikatoren durch Veränderung der Trainingsstichproben
OOB Treffergenauigkeit	P	Schätzung der Treffergenauigkeit, gemessen an den out-of-bag Instanzen der jeweiligen Decision Trees	Beurteilung der Treffergenauigkeit noch vor Evaluierung auf einem Validierungs- oder Testdatenset
Alle (Hyper-)Parameter von Decision Trees	H/P	Die (Hyper-)Parameter der Basisklassifikatoren werden über den Random Forest zentral definiert	Steigerung der Trefferrate des Random Forests durch Optimierung der Hyperparameter seiner Decision Trees

2.6 Erfolgsmessung von Klassifikatoren

Nach der Erstellung eines oder mehrerer Klassifikatoren ist häufig die Güte des Modells von Interesse, um die Anwendbarkeit in der Realität abzuschätzen. Diese Güte wird beispielsweise mit der Fehlerrate gemessen. Die Fehlerraten werden dann verglichen, um den erwartungsgemäß genauesten Klassifikator für einen gegebenen Anwendungsfall zu identifizieren. Außerdem ist die erwartete Fehlerrate auf neuen Daten außerhalb von $\chi_{training}$ häufig von Interesse. Das ist insbesondere der Fall, wenn es für die Fehlerrate eine – selbstaufgeriegte oder fremdbestimmte – harte Obergrenze von $p\%$ gibt, die nachgewiesen werden muss,

bevor der Klassifikator in der Realität angewendet werden darf.

Das Training, die Validierung und das Testen erfolgen jeweils auf den separaten Datensets χ_{training} , $\chi_{\text{validierung}}$ und χ_{test} , wobei gilt

$$\chi_{\text{training}} \cap \chi_{\text{validierung}} = \emptyset; \chi_{\text{validierung}} \cap \chi_{\text{test}} = \emptyset \text{ und } \chi_{\text{training}} \cap \chi_{\text{test}} = \emptyset. \quad (2.19)$$

χ_{training} dient der Optimierung der Parameter eines Klassifikators, $\chi_{\text{validierung}}$ dem Tuning von Hyperparametern und χ_{test} der Erhebung des erwarteten Fehlers auf neuen Daten. Jedes dieser Datensets soll die Grundgesamtheit, aus der die Stichproben stammen, repräsentativ darstellen. Alpaydin (2008) schlägt vor, ein Drittel der vorhandenen Daten für χ_{test} zu verwenden und die anderen zwei Drittel auf χ_{training} und $\chi_{\text{validierung}}$ zu verteilen. Für die Einteilung zwischen χ_{training} und $\chi_{\text{validierung}}$ gibt es mehrere Varianten. Eine davon ist die Kreuzvalidierung.

In der Kreuzvalidierung werden aus χ k Stichproben, $\{\chi_1, \dots, \chi_k\}$ mit $|\chi_1| = \dots = |\chi_k|$ generiert. Dann erfolgen k Durchläufe, genannt Folds, wobei der Klassifikator in *Fold_i* auf $\chi_1 \cup \dots \cup \chi_{i-1} \cup \chi_{i+1} \cup \dots \cup \chi_k$ trainiert wird und auf χ_i getestet wird. Alpaydin (2008) merkt an, dass für jede Teilstichprobe χ_i die relative Häufigkeit jeder Klasse y gleich sein sollte, und, dass diese wiederum den relativen Häufigkeiten von y in χ gleichen sollten. Dieser Vorgang nennt sich Stratifizierung. Ein Spezialfall der Kreuzvalidierung ist die Leave-One-Out Methode (Alpaydin, 2008). Dabei wird χ in $k = n$ Stichproben aufgeteilt, mit $n = |\chi_{\text{training}} \cup \chi_{\text{validierung}}|$. Jede Trainingsstichprobe χ_i besteht also aus einer einzigen Instanz. In diesem Spezialfall ist Stratifizierung nicht möglich. Die Leave-One-Out Methode wird zum Beispiel in der medizinischen Diagnostik verwendet, wenn nur sehr wenige relevante Daten vorhanden sind (Alpaydin, 2008). Eine weitere Variante zur Einteilung der Datensets χ_{training} , $\chi_{\text{validierung}}$ und χ_{test} ist die Bootstrap Methode. Diese wurde bereits in Unterabschnitt 2.4.2 behandelt.

Um die Güte eines Klassifikators zu bestimmen und diesen mit anderen zu vergleichen, bieten sich verschiedene Kennzahlen an. Jede Klassifizierung fällt zunächst in eines der vier Felder aus der Konfusionsmatrix in Tabelle 2.3.

Tabelle 2.3 Aufbau einer Konfusionsmatrix

		\hat{y} (Ergebnis der Klassifikation)	
		1	0
y	1	WP (Wahre Positive)	FN (Falsche Negative)
	0	FP (Falsche Positive)	WN (Wahre Negative)

Die Fehlerrate E berechnet sich dann mit

$$E = \frac{|FP| + |FN|}{N}, \quad (2.20)$$

wobei $N = |WP| + |FP| + |FN| + |WN|$.

Im Falle von stark verzerrten Datensets, wenn die Klassen also ungleich verteilt sind, eignet sich diese simple Fehlerrate alleine nicht zur Erfolgsmessung. Dann bieten sich

weitergehende Metriken aus der Konfusionsmatrix an, wie die Precision, der Recall und das F-Maß (Gron, 2017). Precision misst den Erfolg des Klassifikators auf den positiven Instanzen und ergibt sich aus

$$Precision = \frac{WP}{WP + FP}. \quad (2.21)$$

Precision alleine reicht jedoch nicht aus. Ein Klassifikator könnte von n wahren Positiven nur einen einzigen, sicheren als positiv bestimmen, wobei $1 << n$. Damit wäre die Precision $\frac{1}{(1+0)} = 100\%$, jedoch zu Lasten von $n - 1$ falschen Negativen. Um diesen Trade-Off zu erkennen, bietet sich der Recall als weitere Metrik an. Dieser folgt aus

$$Recall = \frac{WP}{WP + FN}. \quad (2.22)$$

Das F-Maß kombiniert schließlich Precision und Recall in Form des harmonischen Durchschnitts (Gron, 2017)

$$F\text{-Maß} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (2.23)$$

Das F-Maß bestraft zwar Klassifikatoren, die sehr ungleiche Werte für Precision und Recall erzielen. Gleichzeitig bevorzugt es jedoch Klassifikatoren, die für Precision und Recall ähnliche Werte aufweisen (Alpaydin, 2008). Das ist nicht immer erstrebenswert. Je nach Anwendungsfall kann entweder Precision oder Recall relevanter sein (Russell u. Norvig, 2009). Handelt es sich zum Beispiel um die Klassifikation zur Bombendetektion, so würde man vermutlich Fehlalarme einer entgangenen Bombe vorziehen. Dann wäre Recall wichtiger als Precision. Allgemein geht eine höhere Precision mit niedrigerem Recall einher, und vice versa (Gron, 2017). Dieser Precision-Recall Trade-Off ist eine zentrale Fragestellung im Machine Learning und ist für jeden Anwendungsfall einzeln zu lösen.

Es sei erwähnt, dass die Konfusionsmatrix nicht das einzige Kriterium zur Erfolgsmessung eines Klassifikators sein sollte. Turney (2000) ergänzt zum Beispiel die Speicher- und Laufzeit-Komplexität sowie die Interpretierbarkeit der Ergebnisse als weitere Bewertungskriterien, die in der Forschung noch nicht genug Beachtung gefunden hätten.

2.7 Underfitting, Overfitting und der Curse of Dimensionality

Um die bestmögliche Generalisierung des Klassifikators zu erreichen, vergleicht man die Komplexität der erlernten Funktion f mit der den Daten zugrundeliegenden Funktion (Alpaydin, 2008). Ist die Komplexität von f niedriger als die der approximierten Funktion, wird dies als Underfitting bezeichnet. Dies ist zum Beispiel der Fall, wenn man versucht, eine Gerade auf einen Datensatz anzupassen, der von einem Polynom dritten Grades stammt (Alpaydin, 2008). Die Erhöhung der Komplexität führt bei Underfitting zu einer Verbesserung der Vorhersagegenauigkeit beziehungsweise zu einer Reduktion des Vorhersagefehlers. Ist die Komplexität von f hingegen höher als die der approximierten Funktion, so wird dies als Overfitting bezeichnet. Ein Beispiel ist ein nicht-kontrollierter Decision Tree, der für jede Instanz einen eigenen Blattknoten anlegt.

Es folgt die Definition von Overfitting. Dabei bezeichnet $E(\cdot)$ die Fehlerfunktion. Angenommen, der Lernalgorithmus zieht die Funktion $f_1(X)$ aus F einer Funktion $f_2(X)$ vor, weil auf $\chi_{training}$ gilt:

$$E(f_1(x)) < E(f_2(X)). \quad (2.24)$$

Gleichzeitig aber gilt auf der gesamten Verteilung, aus der $\chi_{training}$ stammt:

$$E(f_1(X)) > E(f_2(X)). \quad (2.25)$$

Dann wird die Funktion $f_1(X)$ als overfitted bezeichnet (Mitchell, 1997).

Was genau die Komplexität ausmacht, hängt vom betrachteten Klassifikator ab. In Decision Trees leitet sich die Komplexität aus der Anzahl an Knoten und Blättern ab. Eine große Anzahl an Knoten und Blättern bedeutet eine hohe Komplexität, was wiederum zu Overfitting führen kann, und vice versa. Für Decision Tree Ensembles, wie dem Random Forest, kommt zusätzlich noch die Anzahl der einzelnen Basisklassifikatoren als Faktor für die Komplexität hinzu. Hierbei ist anzumerken, dass eine größere Anzahl an Basisklassifikatoren aber nicht unbedingt zu Overfitting führt. Random Forests profitieren – im Widerspruch zu Ockhams Rasiermesser – von einer steigenden Anzahl an Basisklassifikatoren, solange diese hinreichend unabhängig voneinander sind (Gron, 2017).

Eine weitere, allgemeine Herausforderung – unabhängig vom konkreten Anwendungsfall – im Machine Learning ist der Curse of Dimensionality. Dieses Phänomen bezeichnet die exponentielle Steigung der benötigten Trainingsinstanzen, die bei Erhöhung der Dimensionen, oder der Features, nötig ist (Verleysen u. François, 2005). Wenn beispielsweise ein Klassifikator in einer Dimension mit 10 Instanzen trainiert wird, so sind bei zwei Dimensionen 100 und bei drei Dimensionen 1.000 Trainingsinstanzen nötig, um den gleichen Lernerfolg zu erzielen (Verleysen u. François, 2005). Obwohl Random Forests dazu in der Lage sind, den Curse of Dimensionality zu reduzieren, sind diese dennoch von einer steigenden Anzahl an Features negativ betroffen, da sich die Trainingszeit verlängert (Li, 2016). Li (2016) schlug einen parallelisierten Random Forest vor und zeigte, dass dieser, verglichen mit einem nicht-parallelisierten Random Forest, ein besseres Ergebnis sowie eine um 40% verkürzte Rechenzeit erreicht.

2.8 Besonderheiten bei der Klassifikation von Zeitreihen

Für den Spezialfall der Zeitreihen-Klassifikation bieten sich angepasste Methoden für die Messung der Fehlerrate an. Es ist zum Beispiel möglich, dass zwischen den Instanzen aus χ zeitliche Abhängigkeiten bestehen. So könnte in einer Stichprobe über die Jahre j_1 bis j_n ein Muster ab Jahr j_s auftreten, wobei $j_1 < j_s < j_n$. Ein Beispiel ist die Einführung eines neuen Gesetzes in Jahr j_s , das die betrachtete Zeitreihe ab dem Folgejahr in Jahr j_{s+1} beeinflusst. In diesem Fall sollte der Klassifikator das Muster nur auf Instanzen aus den Jahren $[j_s, j_n]$, nicht jedoch aus der Vorzeit $[j_1, j_s]$ anwenden.

Ein Ansatz, um zeitlich-bedingte Muster zu entdecken, ist die Blocked Form Cross-Validation, auch bekannt als Time Series Cross-Validation. Bergmeir u. Benitez (2011) haben empirisch ermittelt, dass diese Methode für Zeitreihen-Klassifikatoren genauere Ergebnisse erzielt als herkömmliche Cross-Validation Varianten, wie die bereits erwähnte Kreuzvalidierung. Der schematische Ablauf ist in Abbildung 2.2 dargestellt. Ausgehend von

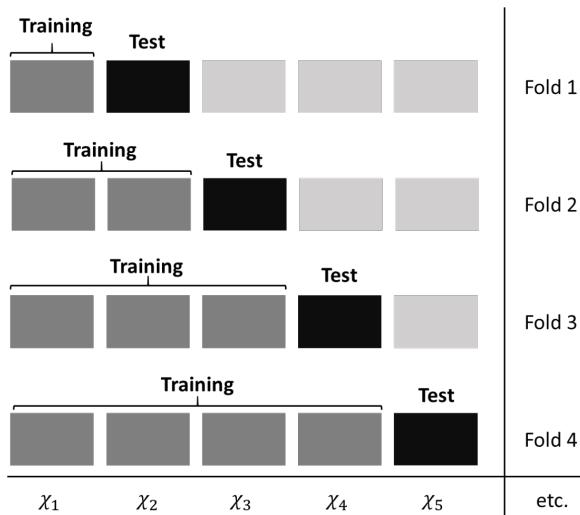


Abbildung 2.2 Schematischer Ablauf einer Time Series Cross-Validation in den ersten vier Iterationen

der Trainingsstichprobe $\chi_{training}$ bildet die Time Series Cross-Validation k Teilstichproben, $\{\chi_1, \dots, \chi_k\}$, wobei $|\chi_1| = \dots = |\chi_k|$. Dann erfolgen $i = k - 1$ Epochen, wobei der Klassifikator in $Epoche_i$ jeweils auf $\chi_1 \cup \dots \cup \chi_i$ trainiert wird und auf χ_{i+1} getestet wird. Alternativ kann $\chi_i \cup \dots \cup \chi_k$ als Testset verwendet werden. Eine weitere Variation ist es, das Modell nicht auf den Daten aller vorangegangener Zeitabschnitte, sondern lediglich auf jenen des letzten Zeitabschnittes, also auf χ_i , zu trainieren (Cerqueira u. a., 2019).

2.9 Random Walk Theorie und Markteffizienzhypothese

Da sich diese Arbeit mit Anwendungsfällen aus der Finanzdomäne beschäftigt, folgen zwei grundlegende Theorien aus diesem Gebiet. Nach der Random Walk Theorie (Fama, 1965) sind sämtliche Analysen zur Vorhersage von Aktienkursen wertlos. Das gilt sowohl für die technische Analyse, die annimmt, dass es in Aktienkursen sich wiederholende Muster gibt, als auch für die Fundamentalanalyse, die den intrinsischen Wert eines Unternehmens anhand von Finanzkennzahlen bewertet (Fama, 1965). Die technische Analyse bezieht sich ausschließlich auf die Zeitreihen eines Wertpapiers. Kennzahlen über die prozentuale oder absolute Veränderung des Kurses sowie andere statistische Werte bilden dabei die Grundlage für Kauf- und Verkaufentscheidungen. Ein Beispiel aus der technischen Analyse ist der gleitende Durchschnitt der Schlusswerte der letzten d Tage, wobei d beliebig variiert wird. Die Fundamentalanalyse hingegen basiert auf den Finanzkennzahlen des Emittenten, das heißt des herausgebenden Unternehmens des Wertpapiers, und nicht auf dem Kurs des Wertpapiers selbst. Beispiele für solche Kennzahlen sind der Umsatz, der Jahresüberschuss oder auch die Eigenkapitalrendite des Unternehmens. Ebenso betrachtet die Fundamentalanalyse ökonomische Faktoren wie den Leitzins, gesellschaftliche Trends oder das Management des Emittenten.

Analysen basieren per Definition stets auf Informationen, die den Marktteilnehmern bekannt sind. Nach der Markteffizienzhypothese sind all diese Informationen jedoch bereits im Preis des Wertpapiers abgebildet. Dann kann die Analyse der Informationen keine

überproportionalen Renditen ermöglichen. Nach Fama (1965) ist ein effizienter Markt ein Markt, in dem sehr viele rationale, Gewinn-maximierende Akteure konkurrieren, wobei jeder von diesen Zugriff auf alle Informationen hat und versucht, die Kursbewegungen gewinnbringend vorherzusagen. Somit pendelt sich der Preis eines Wertpapiers in einem Gleichgewicht ein und spiegelt in einem effizienten Markt alle vorhandenen Informationen wieder. Sowohl Ereignisse aus der Vergangenheit als auch zukünftig erwartete Ereignisse, über die sich die Marktakteure einig sind, bestimmen die Preisbildung. Der Marktpreis stimmt also mit dem inneren Wert des Wertpapiers überein. In einem solchen Markt "hat eine Zeitreihe von Aktienkursen kein Gedächtnis" (Fama, 1965); es ist nicht möglich von historischen Beobachtungen auf neue Kursbewegungen zu schließen. Erkennbare, gewinnbringende Muster würden von der Masse der rationalen Akteure sofort ausgenutzt, bis sie im Marktpreis berücksichtigt und somit nutzlos wären (Lendasse u. a., 2000). Also können nur neue Informationen zu einer Änderung des inneren Preises eines Wertpapiers führen. Und neue Informationen können per Definition nicht vorhergesagt werden (Lendasse u. a., 2000).

In der Realität treffen einige Annahmen der Markteffizienzhypothese allerdings nicht immer zu. Zum einen handeln die Akteure am Aktienmarkt, also Käufer und Verkäufer, nicht immer rational. Auch wenn Computer einen wachsenden Teil der Aktienkäufe und -Verkäufe tätigen, sind emotional geleitete Menschen am Markt aktiv. Zudem hat nicht jeder Marktteilnehmer die gleichen Informationen. Zum Beispiel nutzen die Akteure unterschiedliche Datenquellen, die Informationen verzerren können oder zu verschiedenen Zeitpunkten veröffentlichen. Selbst wenn die Informationsbasis aller Akteure gleich wäre, würde viele von diesen – aufgrund der Einzigartigkeit der Erfahrungen und Fähigkeiten eines jeden Menschen – zu unterschiedlichen Ergebnissen kommen.

Neben den menschlichen Abweichungen vom optimalen Handeln ist es möglich, dass die am Markt teilnehmenden Computerprogramme selbst, durch ihre automatischen Kauf- und Verkaufsaktionen bei Unter- oder Überschreitung gewisser Preispunkte, Muster in den Aktienkursen erzeugen. Dann könnte ein Klassifikator, wie der Decision Tree, eben diese Muster erkennen und Regeln bilden, um sie systematisch auszunutzen und überproportionale Renditen zu erzielen.

2.10 Relevante Literatur

In den letzten Jahren haben Forscher die Anwendung von Machine Learning und speziell von Random Forest Modellen zur Vorhersage von Aktienkursen unter verschiedenen Ansätzen untersucht. Es folgt ein Überblick relevanter Veröffentlichungen mit deren Ansätzen, Vorgehen und Ergebnissen. Dabei werden auch bisher nicht- oder unzureichend erforschte Fragen identifiziert.

Sadia u. a. (2019) haben Random Forests und Support Vector Machines verglichen und deren Vorhersagegenauigkeiten von Aktienkursen auf einem Datensatz von Kaggle gemessen. Insgesamt lagen 121.608 Datensätze verschiedener Aktien vor, die bereinigt und im Verhältnis 80% zu 20% in Trainings- und Testset aufgeteilt wurden. Das Trainingsset wiederum wurde mittels Cross-Validation aufgeteilt, um die Hyperparameter zu optimieren. Mit

einer Genauigkeit von 0,808 hat der Random Forest ein besseres Ergebnis erzielt als die State Vector Machine mit 0,787. Da die Chronologie der Daten bei der Cross-Validation nicht eingehalten wird, sind in dieser Arbeit mögliche zeitliche Abhängigkeiten nicht berücksichtigt worden. Außerdem ist die 80% zu 20% Aufteilung nur einmalig erfolgt. Im Gegensatz zu Cross-Validation Ansätzen liegt hier eine lokale Verzerrung vor, da kein Mittelwert über verschiedene Aufteilungen berechnet wurde. Die Anwendung der Time Series Cross-Validation reduziert die beiden genannten Schwächen und kann somit zu aussagekräftigeren Ergebnissen führen.

Alavi u. a. (2015) haben Random Forests mit State Vector Machines und K-Nearest Neighbor Modellen verglichen. Als Datengrundlage dienten Zeitreihen iranischer Aktien im Zeitraum von 2002 bis 2012. Als zusätzliche Features, neben den Aktienkursen selbst, wurden Indikatoren aus der technischen Aktienanalyse miteinbezogen. Die Untersuchungen wurden in MATLAB durchgeführt. Nach dem Hyperparameter-Tuning erreichte der Random Forest mit 0,919 das beste F-Maß, gefolgt von State Vector Machines (0,860) und K-Nearest Neighbors (0,820). Zur Erfolgsmessung haben die Autoren eine einmalige Aufteilung in Trainings- und Testset vorgenommen; ein auf Zeitreihen spezialisiertes Verfahren hat nicht stattgefunden. Auch die Auswirkungen der zusätzlichen technischen Features wurde nicht eruiert.

Pasupulety u. a. (2019) haben einen Baum-basierten Klassifikator mit Features zur Erfassung der öffentlichen Wahrnehmung einer Aktie (Sentiment Analysis) angereichert, um Aktienkurse des indischen IT-Unternehmens Infosys Limited vorherzusagen. Dabei konnten sie jedoch nur eine vernachlässigbare Verbesserung der Genauigkeit feststellen. Die Umsetzung erfolgte mit der Scikit-learn Bibliothek in Python. Die Erfolgsmessung fand dabei nach der Time Series Cross-Validation statt.

Eine Alternative zur binären Klassifikation auf den Aktienmärkten, mit den Klassen "Aktie steigt" und "Aktie fällt", schlügen Lohrmann u. Luukka (2019) vor. Mit den vier Klassen "stark positiv", "schwach positiv", "schwach negativ" und "stark negativ" konnten sie bessere Ergebnisse erzielen, als mit zwei Klassen. Aktienkurse zwischen 2010 und 2018 von der Yahoo Finance API waren dabei die Datengrundlage. Der Random Forest Klassifikator erreichte in den MATLAB-Simulationen die höchste Genauigkeit. Eine Untersuchung verschiedener Zeithorizonte fand nicht statt.

Khaidem u. a. (2016) haben bei der Klassifikation nach Zeithorizont unterschieden und erschlossen daraus, dass längere Zeithorizonte zu einer höheren Treffergenauigkeit führen. Für Datensätze von Apple, Microsoft und Samsung berichten die Autoren Treffergenauigkeiten zwischen 85% und 95% für einen Horizont von zehn Tagen, was eine Steigerung gegenüber der Bezugsliteratur (Di, 2014) darstellt, in der Werte zwischen 70% und 80% erreicht wurden. Hierbei wurden die Ergebnisse auf den verschiedenen Datensätze nicht miteinander verglichen. Es ist jedoch zu vermuten, dass sich die Klassifikatoren hinsichtlich ihrer Genauigkeit zwischen den einzelnen Aktien erheblich unterscheiden. Ebenso sollte untersucht werden, wie sich die Klassenverteilungen zwischen den Zeithorizonten unterscheiden. Denn sehr einseitige Klassenverteilungen verzerrn den Klassifikator, sodass dieser zwar die überrepräsentierte Klasse erfolgreich vorhersagen kann, nicht aber die unterrepräsentierte. Das wiederum kann die Ergebnisse maßgeblich beeinflussen.

Die Vorhersagegenauigkeit von Machine Learning Modellen kann abhängig vom Zeithorizont stark schwanken. So ist es möglich, dass es für die Vorhersagen ein Jahr in die Zukunft erkennbare Muster gibt, während Vorhersagen einen Tag in die Zukunft dem Zufall unterliegen. Einige der genannten Veröffentlichungen haben den Zeithorizont der Vorhersage gar nicht beachtet, andere haben sich auf maximal drei Monate beschränkt, dafür jedoch die kurzen Horizonte von wenigen Tagen vernachlässigt. Vorhersagen über mehr als drei Monate hinweg wurden nicht betrachtet. Auch ist noch unerforscht, inwiefern sich die Gewichte der Features mit den Zeithorizonten verändern. So ist es vorstellbar, dass zum Beispiel ein Decision Tree abhängig vom Zeithorizont jene Features stärker gewichtet, die einen entsprechend langen oder kurzen Zeitraum erfassen. Soll das Modell die Kursentwicklung ein Jahr in die Zukunft vorhersagen, könnte es langfristige Features, die Aussagen über das letzte Jahr treffen, den kurzfristigen Features, über die letzten Tage oder Wochen, vorziehen.

Zusammenfassend lässt sich feststellen, dass der Vergleich der Modelle zwischen den Aktien – und nicht nur im Gesamtmarkt –, der Einfluss von technischen Features und von Hyperparameter-Tuning auf die Vorhersagegenauigkeiten sowie die Veränderung der Gewichtungen der Features und der Klassenverteilungen mit unterschiedlichen Zeithorizonten noch nicht ausreichend erforscht sind. Auch der Vergleich der Klassifikatoren mit einem Dummy ist häufig nicht vorhanden, wodurch ein objektiver Vergleichsmaßstab fehlt und die Aussagekraft einschränkt. Die genannten Lücken stellen die Schwerpunkte im nachfolgenden Forschungsteil dar.

3 Methodik

3.1 Vorgehen

Nachfolgend wird das Vorgehen zur Untersuchung der Forschungsfragen dargelegt. Um die verschiedenen Modelle miteinander zu vergleichen, werden die in Tabelle 3.1 erläuterten fünf Variablen eingeführt.

Tabelle 3.1 Die fünf zentralen Variablen für die Untersuchungen

Variable	Mögliche Werte
Klassifikator	{Dummy Klassifikator, Decision Tree, Random Forest}
Datenset (Tickersymbol der Aktie)	{AAPL, AMZN, CSCO, GE, GOOGL, HP, IBM, INTC, MSFT, WU, XRX}
Feature Extraction	{Ja, Nein}
Zeithorizont	{1d, 5d, 10d, 20d, 65d, 250d} mit den Indizes {0, 1, 2, 3, 4, 5}
Hyperparameter-Tuning	{Ja, Nein}

Die erste Variable ist der Klassifikator selbst. Er nimmt stets eine der drei genannten Ausprägungen – Dummy Klassifikator, Decision Tree oder Random Forest – an. In nachfolgenden Diagrammen werden diese abgekürzt mit Dummy, DT und RF. Die zweite Variable ist das Datenset. Die elf ausgewählten Aktien-Datensets stammen allesamt aus dem Technologie Sektor und werden in Abschnitt 3.3 im Detail vorgestellt. Die dritte Variable ist das Feature Extraction. Um die Auswirkungen der zusätzlichen technischen Features zu erfassen, werden die Klassifikatoren im ersten Durchgang ohne diese trainiert und getestet, und im zweiten Durchgang mit ihnen. Die vierte Variable ist der Zeithorizont, der zwischen einem Handelstag (1d) und einem Handelsjahr (250d) liegt. Die fünfte Variable ist das Hyperparameter-Tuning. Dadurch soll eruiert werden, wie sich die Optimierung der Hyperparameter in verschiedenen Szenarien auf die Treffergenauigkeit des Modells auswirkt. In den Auswertungen werden die Modelle mit Hyperparameter-Tuning als TunedDT (Decision Tree) und TunedRF (Random Forest) abgekürzt. Das allgemeine Vorgehen ist so, dass vier dieser Variablen konstant gehalten werden und die fünfte variiert. Anschließend werden Rückschlüsse gezogen, wie diese fünfte Variable das Ergebnis beeinflusst. Für jede Kombination dieser fünf Variablen wird dasselbe Prozedere angewendet, um sie zu bewerten und mit den anderen Kombinationen zu vergleichen.

Wie in Abbildung 3.1 dargestellt, ist der erste Schritt das Laden eines Datensets, das im Format .CSV (Comma-Separated Values) vorliegt. Eine Vorstellung der ausgewählten Datensets folgt in Abschnitt 3.3. Das Ergebnis ist eine Datenstruktur, die die Zeitreihen eines Aktienkurses enthält. Nach dem Laden stehen für jeden enthaltenen Tag das Datum ("date"), der Name der Aktie ("name"), der Eröffnungspreis ("open"), das Tageshoch ("high"), das

Tagestief ("low"), der Schlusspreis ("closing") und das gehandelte Volumen des Tages ("volume") zur Verfügung. Ein Vorteil von Decision Trees und Random Forests ist es, dass die Werte der Features keine Skalierung benötigen (Gron, 2017). Somit entfällt dieser Schritt. Auch die Auswahl der wichtigsten Features ist, im Gegensatz zu anderen Machine Learning Modellen, hier nicht nötig. Die Bäume selektieren nämlich während ihrer Erstellung automatisch die wichtigsten Features insofern, als deren Aufteilungen anhand jener Features stattfinden, welche die Entropie der Klassen am stärksten reduzieren (siehe Unterabschnitt 2.3.2).

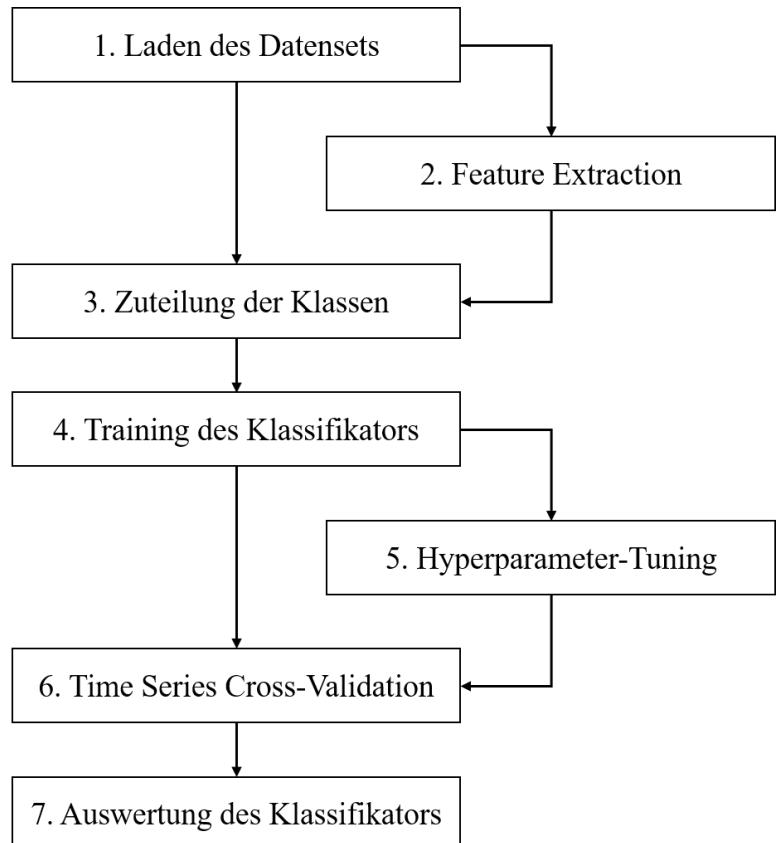


Abbildung 3.1 Vorgehen in der Untersuchung vom Laden des Datensets bis zur Auswertung

Im zweiten Schritt werden weitere Features aus der technischen Analyse berechnet. Zu Untersuchungszwecken kann dieser Schritt übersprungen werden. Einige, in der Literatur weit verbreitete technische Features sind der gleitende Durchschnitt, die Volatilität und das Momentum (Drakopoulou, 2016). Der gleitende Durchschnitt der Schlusspreise, c_i , der letzten k Tage, benannt als $ma_{-}(k)d$, ergibt sich zum Zeitpunkt t aus

$$ma_{-}(k)d = \frac{\sum_{i=t-k+1}^t c_i}{k}. \quad (3.1)$$

Die Volatilität der letzten k Tage, $volatility_{-}(k)d$, zum Zeitpunkt t wird mit der Standardabweichung

$$volatility_{-}(k)d = \sqrt{\frac{\sum_{i=t-k+1}^t (c_i - \bar{x})^2}{k-1}} \quad (3.2)$$

gemessen. Das Momentum der letzten k Tage, $momentum_{\langle k \rangle d}$, bezeichnet die absolute Veränderung der Schlusspreise bis zum Zeitpunkt t :

$$momentum_{\langle k \rangle d} = c_t - c_{t-k}. \quad (3.3)$$

Ebenso wird die prozentuale Veränderung der Schlusspreise der letzten k Tage, $return_past_{\langle k \rangle d}$, als Feature ergänzt:

$$return_past_{\langle k \rangle d} = \frac{c_t}{c_{t-k}} - 1. \quad (3.4)$$

Für k werden jeweils die betrachteten Zeithorizonte h_i aus Tabelle 3.1 eingesetzt. Zuletzt wird das Feature "ohlc_avg" ergänzt, das als Durchschnitt den Eröffnungspreis, das Tageshoch, das Tagestief und den Schlusspreis zusammenfasst. Insgesamt stehen den Klassifikatoren somit 31 Features zur Verfügung; eine Auflistung findet sich in Anhang A.2.

Der dritte Schritt ist die Zuteilung der Klassen. Bisher enthält die Datenstruktur lediglich Features. Um die Auswirkungen unterschiedlicher Klassifikationshorizonte zu untersuchen, wird pro Zeithorizont h_i eine eigene Klassenspalte erstellt. Dazu wird zuerst berechnet, ob der Schlusspreis in den nächsten h_i Tagen steigt oder fällt. Falls der Kurs steigt, wird der Instanz die Klasse 1 (positiv) zugeteilt. Andernfalls lautet die Klasse 0 (negativ). Nach dieser Logik zählt auch ein konstanter Kurs, dessen Werte zu Beginn und am Ende von h_i identisch sind, zur negativen Klasse. Der Klassifikator soll in diesem Fall also keine Kaufempfehlung abgeben. Zeithorizonte werden stets in Handelstagen angegeben, weshalb der maximale Horizont über ein Jahr 250 Tage besitzt. So gibt zum Beispiel die Klasse $class_10d$ der Instanz X vom 01.01.2015 an, ob der Kurs der Aktie bis zum 15.01.2015 steigt oder fällt. Nachdem die Klassen erstellt sind, werden alle Spalten mit Zwischenergebnissen, die zur Erstellung dieser Klassen verwendet wurden, aus der Datenstruktur entfernt. Dadurch wird verhindert, dass der Klassifikator von illegal-voraussehenden Features profitiert, die die Erfolgsmessung positiv verfälschen würden.

Nach der Klassendefinition folgt das Training des Klassifikators. Das Training erfolgt auf jenen Daten, die in der aktuellen Epoche als Trainingsdaten indiziert sind. Pro vergangener Epoche stehen ungefähr 126, bzw. 76, zusätzliche Trainingsdaten zur Verfügung, die in der jeweils letzten Epoche noch als Testset dienten. Das bedeutet, dass in der ersten Epoche die wenigsten, und in der letzten Epoche die meisten Trainingsdaten vorhanden sind. Es werden in dieser Arbeit drei Klassifikatoren verglichen. Ein Dummy Klassifikator dient als Vergleichsmaßstab. Um einen Klassifikator sinnvoll in der Realität einsetzen zu können, muss er eine signifikant höhere Trefferrate als der Dummy erreichen. Der Dummy betrachtet nur die Verteilung der Klassen in den Trainingsdaten, nicht jedoch die sonstigen Features. Dann errechnet er die Wahrscheinlichkeit $P(y_i)$ pro Klasse y_i . Bei der Klassifikation unbekannter Instanzen erteilt er jeder unbekannten Instanz dann mit Wahrscheinlichkeit $P(y_i)$ die Klasse y_i . Der Dummy nimmt an, dass die Klassenverteilung auf den Trainingsdaten stets mit der Verteilung auf den Testdaten übereinstimmt. Ausgehend von den Treffergenauigkeiten dieses simplen Klassifikators lassen sich andere Modelle vergleichen. Die weiteren zwei Klassifikatoren sind der Decision Tree (siehe Abschnitt 2.3) und der Random Forest (siehe Abschnitt 2.5). Für alle Klassifikatoren werden die Implementierungen von Scikit-learn verwendet. Der Random Forest wird, aufgrund begrenzter Ressourcen, standardmäßig mit fünf Basisklassifikatoren initialisiert.

Der fünfte Schritt ist das Hyperparameter-Tuning, mit dem Decision Trees und Random Forests verbessert werden können. Dieser Schritt ist optional. Für jeden der beiden Klassifikatoren werden zuerst zentrale Hyperparameter aus den Tabellen 2.1 und 2.2 ausgewählt. Dann werden für diese festgelegte Werte auf einem Validierungsset ausprobiert und verglichen, um die beste Kombination der Hyperparameter-Werte zu ermitteln. Wie auch bei der Bewertung der Klassifikatoren wird die Güte hier mit dem F-Maß bestimmt. Für den Hyperparameter der maximalen Anzahl an betrachteten Features ist die Wurzel der Anzahl an Trainingsinstanzen ein gängiger Standardwert (Probst u. a., 2019). In dieser Arbeit werden beim Hyperparameter-Tuning Werte bis maximal zu diesem Standardwert verglichen, um das Overfitting zu reduzieren. Auch für die anderen Hyperparameter der maximalen Tiefe, der minimalen Anzahl an Instanzen für die Aufteilung und der minimalen Anzahl an Instanzen für ein neues Blatt werden die zu vergleichenden Werte bis maximal zu diesem Standardwert gewählt. Die Anzahl der Basisklassifikatoren im Random Forest stellt eine Ausnahme dar, da sie nicht gleichermaßen durch Tuning optimierbar ist, sondern lediglich hinreichend hoch gesetzt werden muss (Probst u. a., 2019). Deshalb wird dieser Hyperparameter separat gesetzt.

Neben dieser sogenannten Grid Search, bei der die zu vergleichenden Werte im Vorhinein vom Anwender festzulegen sind, gibt es auch noch eine zufallsgesteuerte Alternative, die Randomized Grid Search. Dann werden die zu vergleichenden Hyperparameter-Werte nicht explizit vom Anwender angegeben. Stattdessen werden zufällige Kombinationen solange betrachtet, bis eine vorher festgelegte maximale Anzahl zu vergleichender Kombinationen erreicht ist (Feurer u. Hutter, 2019). Die Randomized Grid Search ist besonders erfolgreich, wenn es nur wenige Features gibt, die mit sehr hohen Gewichtungen die wichtigsten sind (Feurer u. Hutter, 2019). Das Hyperparameter-Tuning wird in jeder der zehn Epochen aus der Time Series Cross-Validation separat ausgeführt. Ein einmaliges Tuning auf dem gesamten Datenset ist nicht möglich, da der Klassifikator sonst bereits Testdaten gesehen hätte und die Trefferrate positiv verfälscht würde.

Im sechsten Schritt findet die Time Series Cross-Validation statt. Diese ist für die vorliegenden Datensets besser geeignet als andere Cross-Validation Ansätze, da sie die Chronologie der Instanzen beibehält (siehe Abschnitt 2.8). Nicht nur technisch, sondern auch fachlich entspricht das der Problemstellung. Denn der Anwender des Klassifikators trainiert diesen auf historischen Daten, bis zum aktuellen Zeitpunkt, und wendet die Klassifikation dann auf den Zeithorizont h_i an, um eine Investitionsentscheidung zu treffen. In der realen Anwendung stehen nicht, wie es zum Beispiel bei der k-fold Cross-Validation der Fall wäre, Datensätze aus der Zukunft, die zeitlich hinter h_i liegen, zur Verfügung. Deshalb findet in dieser Arbeit die Time Series Cross-Validation Anwendung. Für die Anzahl der Epochen wird 10 gewählt; dadurch bestehen die Testsets pro Datensatz aus näherungsweise $\frac{1}{10} * 1.259 = 126$ Instanzen. Werden allerdings die ersten und die letzten 250 Instanzen entfernt, um die 250d-Features für alle überbleibenden Instanzen korrekt berechnen zu können, so stehen pro Testset circa $\frac{1}{10} * (1.259 - 500) = \frac{1}{10} * (759) = 76$ Beispiele zur Verfügung. Mehr als zehn Epochen würden zu sehr kleinen oder zu kleinen Testsets führen, und weniger als zehn Epochen würden die Anzahl der Ergebnisse stärker einschränken und lokale Verzerrungen wahrscheinlicher machen.

Der letzte Schritt ist schließlich die Auswertung des Klassifikators in der jeweiligen Kombination der fünf Untersuchungsvariablen. Die Vorhersagen des trainierten, und gegebenenfalls optimierten, Klassifikators für das unbekannte Testset werden mit den richtigen Klassen

anhand von Konfusionsmetriken verglichen. Die Erfolgsmessung durch Precision, Recall und das F-Maß erfolgt wie in Abschnitt 2.6 beschrieben. Ebenso werden die Gewichtungen der einzelnen Features, genannt Feature Importances, analysiert. Im Falle von Decision Trees und Random Forests ergeben sich diese Feature Importances aus der relativen Reduktion eines Unreinheitsmaßes, die der Baum durch Aufteilungen mittels des jeweiligen Features erreicht.

Diese sieben Schritte werden mehrmals durchgeführt, um die Ergebnisse je nach Kombination der Untersuchungsvariablen zu vergleichen und somit die Forschungsfragen zu beantworten. Für jedes Datenset (11 Möglichkeiten), mit oder ohne Feature Extraction (2 Möglichkeiten), mit einer der Klassen (6 Möglichkeiten), werden drei Klassifikatoren (3 Möglichkeiten), mit oder ohne Hyperparameter-Tuning (2 Möglichkeiten), trainiert und getestet. Insgesamt ergeben sich theoretisch $11 \times 2 \times 6 \times 3 \times 2 = 792$ mögliche Kombinationen. Von diesen werden im Ergebnisteil allerdings nur die für die Forschungsfragen relevantesten Kombinationen direkt miteinander verglichen. Ebenso werden Durchschnitte über einige der fünf Dimensionen, wie beispielsweise über alle elf Datensets, berechnet, um übersichtliche und aussagekräftige Ergebnisse zu erhalten.

3.2 Tool-Stack und Bibliotheken

Die Untersuchungen in dieser Arbeit werden in einem Jupyter Notebook umgesetzt. Jupyter ist ein Open-Source Projekt mit dem Ziel, eine interaktive, webbasierte und sprachenunabhängige Entwicklungsumgebung für Data Science-Anwendungen bereitzustellen. Ein Vorteil von Jupyter ist, dass sich Code, Graphiken sowie Markup-Texte in einer gemeinsamen Datei befinden, und man als Anwender die Ergebnisse somit zentral vorliegen hat. Die Arbeit selbst ist mit der LaTeX-Software verfasst. Die Programmierung findet auf Azure Notebooks, einem Microsoft-Service für gehostete Jupyter Notebooks, statt. Als Kernel dient Python 3.6.6 aus der Anaconda Distribution, die bereits viele Machine Learning-relevante Pakete enthält. Das öffentliche GitHub Repository <https://github.com/feschu/BSc-Thesis-ML> enthält das Jupyter Notebook, die Datensets, die Graphiken sowie Auswertungen. Die Software ist unter der MIT-Lizenz veröffentlicht.

Diese Arbeit nutzt die Pakete Scikit-learn, pandas, Numpy, Matplotlib, Seaborn und Graphviz. Die Open-Source Bibliothek Scikit-learn stellt Implementierungen für eine Vielzahl von Machine Learning Algorithmen, wie dem Decision Tree und dem Random Forest, bereit (Pedregosa u. a., 2011). Auch nützliche Methoden rund um den Machine Learning Prozess deckt Scikit-learn ab, so zum Beispiel die Aufteilung von Daten in Training- und Testsets oder das Hyperparameter-Tuning. Zum Zweck der Reproduzierbarkeit werden Zufallszahlen stets mit dem Wert 42 initialisiert. pandas bietet Werkzeuge zur Verarbeitung und Analyse von Daten an, wie zum Beispiel das zweidimensionale DataFrame. NumPy liefert eine effiziente Array-Implementierung und wird für mathematische Operationen, zum Beispiel zur Mittelwertberechnung oder zur Sortierung, verwendet. Matplotlib ist eine verbreitete Bibliothek zur Visualisierung in Python, die verschiedene Diagrammtypen unterstützt. Seaborn ist eine Erweiterung von Matplotlib und ist auf attraktive und informative Graphiken spezialisiert. Zur Visualisierung von Decision Trees wird Graphviz verwendet.

3.3 Datengrundlage

Elf Datensets bilden die Grundlage für die Untersuchungen. Es handelt sich um die Aktienkurse der Unternehmen Apple (AAPL), Amazon (AMZN), Cisco (CSCO), General Electric (GE), Google (GOOGL), Hewlett-Packard (HP), IBM (IBM), Intel (INTC), Microsoft (MSFT), Western Union (WU) und Xerox (XRX) im Zeitraum vom 08.02.2013 bis zum 07.02.2018. Pro Aktie stehen somit 1.259 Handelstage zur Verfügung. Bei der Auswahl der Aktien wurde darauf geachtet, dass sie sich in ihrem Verlauf innerhalb der fünf Jahre unterscheiden. So folgte zum Beispiel AMZN einem steigenden Trend, HP einem alternierenden Trend und GE einem fallenden Trend, wie in Abbildung 3.2 zu sehen.

Alle elf Unternehmen sind im Technologie-Sektor angesiedelt, teilweise jedoch mit unterschiedlichen Schwerpunkten. Während Google auf Software spezialisiert ist und seinen Umsatz größtenteils mit Werbung generiert, ist Cisco zum Beispiel auf IT-Infrastruktur fokussiert. Die Datensets stammen aus dem Kaggle-Repository "S&P 500 stock data" (<https://www.kaggle.com/camnugent/sandp500>). Kaggle ist eine Plattform, die Machine Learning Wettbewerbe veranstaltet und es den Nutzern unter anderem ermöglicht, Datensets auszutauschen und die Anwendung von Machine Learning Algorithmen auf diesen zu diskutieren.

Die Abbildung 3.2 zeigt den relativen Verlauf der elf Aktien, ausgehend vom 08.02.2013. Die absoluten Werteverläufe der Aktien sind im Anhang A.1 zu finden.

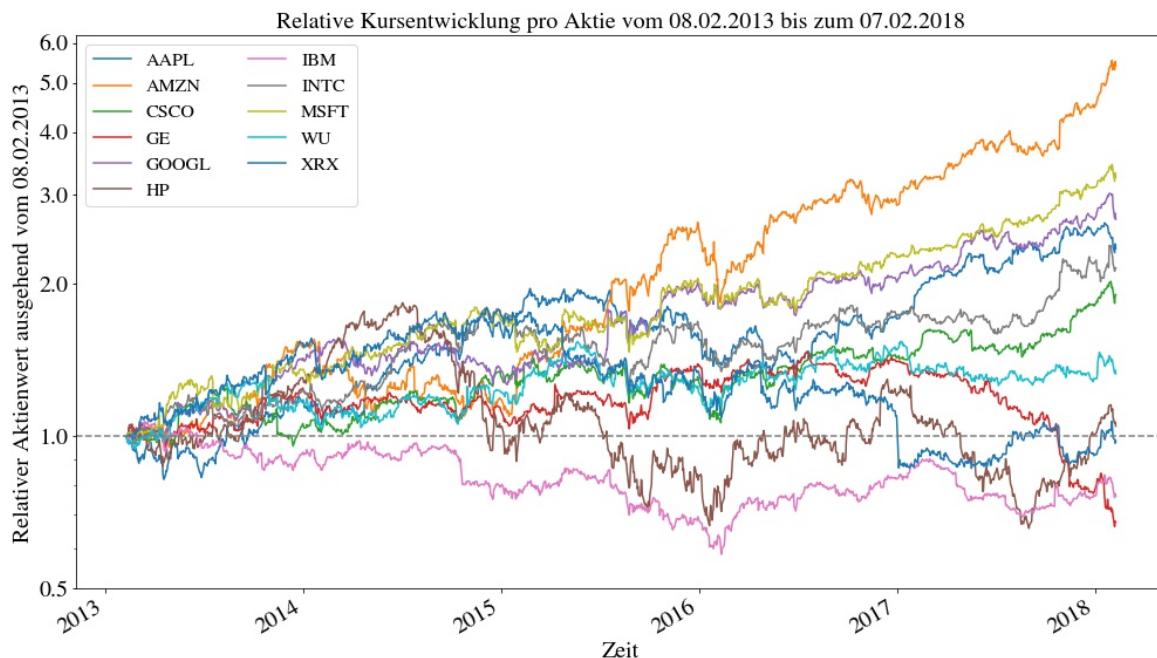


Abbildung 3.2 Relative Kursverläufe der Beispieldatensets ausgehend vom 08.02.2013

Die Aktienkursdaten liegen tageweise für die Werkstage zwischen dem 08.02.2013 und dem 07.02.2018 vor. Pro Tag stehen die Abschnitt 3.1 genannten Features zur Verfügung. Ein beispielhafter Auszug aus dem AAPL-Datenset ist in Tabelle 3.2 zu sehen. Das Datum ist im Format YYYY-MM-DD angegeben. Die Monatsspalte ("month") enthält Ganzzahlen

zwischen 1 und 12, die für die Monate Januar bis Dezember stehen. Diese Spalte wurde aus dem Datum generiert und als zusätzliches Feature eingefügt, damit die Modelle eventuelle zyklische Effekte erkennen können.

Tabelle 3.2 Beispielwerte für die Features aus dem AAPL-Datenset

date	open	high	low	close	volume	name	month	...
2013-02-08	67.7142	68.4014	66.8928	67.8542	158168416	AAPL	2	...
2013-02-11	68.0714	69.2771	67.6071	68.5614	129029425	AAPL	2	...
2013-02-12	68.5014	68.9114	66.8205	66.8428	151829363	AAPL	2	...
2013-02-13	66.7442	67.6628	66.1742	66.7156	118721995	AAPL	2	...
2013-02-14	66.3599	67.3771	66.2885	66.6556	88809154	AAPL	2	...

Anhand der Schlusspreise werden die in Abbildung 3.3 zu sehenden Klassen wie beschrieben pro Zeithorizont berechnet. Es ist zu beachten, dass pro Klassifikation jeweils nur eine einzige Klasse auf einmal betrachtet wird.

Tabelle 3.3 Beispielwerte für die Klassen aus dem AAPL-Datenset

...	class_1d	class_5d	class_10d	class_20d	class_65d	class_250d
...	1	0	0	0	0	1
...	0	0	0	0	0	1
...	0	0	0	0	0	1
...	0	0	0	0	0	1
...	0	0	0	0	0	1

4 Ergebnisse

Nachfolgend findet pro Fragestellung zuerst eine Beschreibung der Ergebnisse statt. Dann werden diese jeweils kritisch analysiert, und mit der Theorie in Bezug gesetzt. Diagramme und Tabellen verschaffen einen schnellen und strukturierten Überblick und unterstreichen die wichtigsten Erkenntnisse. Aufgrund der hohen Anzahl untersuchter Variablen stehen zu viele Ergebnisse bereit, als dass diese im Kapitel eingebunden werden könnten. Nur die zentralen, häufig über mehrere Variablen aggregierenden Diagramme sind zu sehen. Die detaillierteren Ergebnisse, meistens in höherer Granularität wie beispielsweise pro Aktienset, befinden sich im Anhang. Limitierungen der Aussagekraft sollen, wo zutreffend, aufgezeigt werden. Im Anschluss werden weitergehende Ideen sowie aufgekommene Teilfragen präsentiert, die Ausgangspunkte für weitere Forschung darstellen.

4.1 Erfolgsmessung der Klassifikatoren

Der erste Ergebnisabschnitt beschäftigt sich mit der Treffergenauigkeit der Klassifikatoren auf den Datensets, gemessen mit dem F-Maß. Zusätzlich sind die zugrundeliegenden Precision- und Recall-Werte angegeben. Die Tabelle 4.1 zeigt die drei Metriken in Abhängigkeit von dem Zeithorizont für alle betrachteten Klassifikatoren. Die Werte stellen die Durchschnitte über alle elf Datensets dar.

4.1.1 Auswertung der F-Maße

Die erste Beobachtung bezüglich der F-Maße ist, dass sich die Werte für den kürzesten und den längsten Zeithorizont stark unterscheiden. Die F-Maße für den 250-Tage-Horizont liegen deutlich über jenen für den 1-Tages-Horizont. Das trifft auf alle fünf Klassifikatoren zu. Im Schnitt liegen die Genauigkeiten bei 0,827 (250d) bzw. 0,515 (1d), mit einer signifikanten Differenz von 0,312.

Um die Forschungsfrage zu beantworten, ob Decision Trees und Random Forests zur Aktienklassifikation sinnvoll eingesetzt werden können, werden zuerst die zwei Extrempunkte betrachtet: 1d und 250d. Für den kürzesten Horizont erzielen die Decision Tree und Random Forest Klassifikatoren F-Maße von circa 0,5. Sofern die beiden Klassen "Aktie steigt" und "Aktie fällt" gleich wahrscheinlich sind, entsprechen diese F-Maße zufälligem Raten. Verglichen mit dem Ergebnis von 0,531 des Dummy Klassifikators – der sich als Vergleichsmaßstab besser als das zufällige Raten eignet, da er die Verteilungsfunktion der Klassen beachtet – sind die erzielten F-Maße niedriger. Auch wenn die Decision Tree und Random Forest Klassifikatoren mit Tuning bessere F-Maße erreichen als ohne Tuning, ist der Dummy dennoch in den Horizonten 1d bis 20d erfolgreicher. Daraus folgt, dass die Decision Tree und Random Forest Klassifikatoren bei kurzfristigen Horizonten von bis zu 20 Tagen zur Aktenvorhersage nicht sinnvoll einsetzbar sind.

Betrachtet man nun die längerfristigen Horizonte 65d und 250d, so lässt sich feststellen, dass der Dummy Klassifikator allen anderen Klassifikatoren deutlich unterliegt. Im 60-Tage-Horizont sind zwar alle Decision Tree und Random Forest Varianten genauer als der Dummy, teilweise jedoch nicht signifikant. Zum Beispiel ist der Decision Tree um 0,014 besser, oder mit Tuning um 0,042. Der Random Forest übertrifft den Dummy um 0,065, oder mit Tuning um 0,018. Für den 60-Tage-Horizont lässt sich somit keine eindeutige Aussage treffen. Im 250-Tage-Horizont jedoch liegt der Dummy bei 0,742, während die vier komplexeren Klassifikatoren in dem Intervall zwischen 0,835 und 0,862 liegen. Der Random Forest mit Tuning weist das höchste F-Maß von 0,862 auf, um 0,120 höher als der Vergleichsmaßstab. Alle vier komplexeren Modelle übertreffen den Dummy signifikant. Somit sind Decision Tree und Random Forest Klassifikatoren für den 250-Tage-Horizont sinnvoll einsetzbar. Das bedeutet, dass die Modelle dazu in der Lage sind, anhand der Beispieldaten so zu lernen, dass sie bessere Aussagen über die Zukunft treffen als der Dummy Klassifikator.

Tabelle 4.1 Erfolgsmessung pro Klassifikator für die verschiedenen Zeithorizonte

		Durchschnitt über alle Datensets					
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,519	0,531	0,538	0,540	0,582	0,743
	DT	0,514	0,553	0,551	0,540	0,632	0,845
	RF	0,515	0,553	0,571	0,542	0,658	0,885
	TunedDT	0,515	0,547	0,556	0,550	0,637	0,836
	TunedRF	0,521	0,539	0,556	0,550	0,618	0,879
Recall	Dummy	0,543	0,576	0,570	0,589	0,633	0,741
	DT	0,476	0,509	0,463	0,497	0,611	0,828
	RF	0,493	0,505	0,501	0,503	0,686	0,832
	TunedDT	0,544	0,488	0,438	0,508	0,662	0,835
	TunedRF	0,507	0,548	0,507	0,519	0,632	0,845
F-Maß	Dummy	0,531	0,552	0,554	0,563	0,607	0,742
	DT	0,495	0,530	0,503	0,518	0,621	0,836
	RF	0,504	0,528	0,534	0,522	0,672	0,858
	TunedDT	0,529	0,516	0,490	0,528	0,649	0,835
	TunedRF	0,514	0,543	0,530	0,534	0,625	0,862

Um die Veränderungen der F-Maße mit steigendem Horizont miteinander zu vergleichen, sind diese als Linien in Abbildung 4.1 zu sehen. Das F-Maß ist wieder als Durchschnitt über die Datensets berechnet. Es ist bei der Darstellung zu beachten, dass die X-Achse verzerrt ist, denn zwischen den äquidistanten Markierungen liegen zuerst vier Tage, dann fünf Tage, zehn Tage, 45 Tage und schließlich 185 Tage. Diese Verzerrung führt graphisch, vor allem im Abschnitt zwischen 20d und 250d, zu einer höheren Steigung, als es bei einer linearen X-Achse der Fall wäre.

In Abbildung 4.1 ist erkennbar, dass auf kurzfristigen Horizonten bis zu 20d der Dummy Klassifikator die besten Ergebnisse erzielt. Ab 65d jedoch erreicht der Dummy das schlechteste Ergebnis, die anderen vier Klassifikatoren weisen deutlich höhere F-Maße auf. Dabei hat sich diese Arbeit auf maximal 250 Tage beschränkt. Mit größeren Datensets bietet es sich zukünftig an, Horizonte über mehrere Jahre auszuprobieren. Denn hält der beobachtete

positive Trend der F-Maße an, wie in Abbildung 4.1 zu sehen, so könnten sich dadurch zuverlässigere Modelle finden.

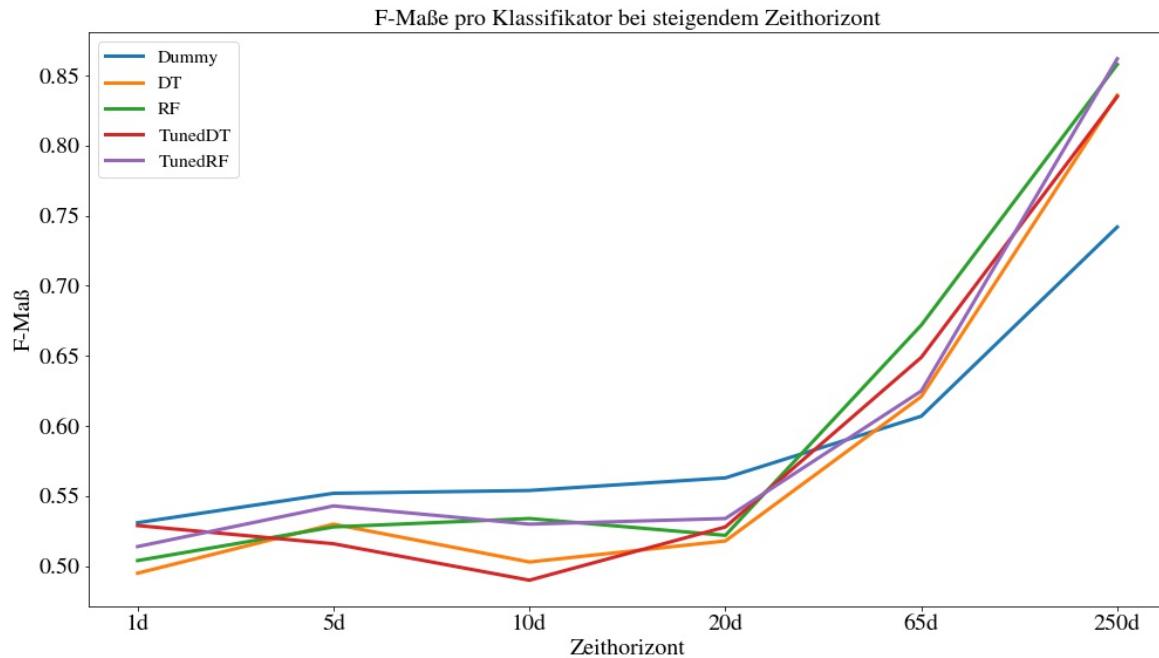


Abbildung 4.1 F-Maße der Klassifikatoren bei zunehmendem Zeithorizont

Zuletzt veranschaulicht die Tabelle 4.1 auch die Unterschiede zwischen den F-Werten von Decision Tree und Random Forest Klassifikatoren ohne Tuning und jenen mit Tuning. Diesbezüglich ist zu sehen, dass das Tuning der Hyperparameter nur teilweise zu besseren Ergebnissen führt. In 7 von 12 Fällen bewirkt das Tuning eine Steigerung des F-Maßes, in den restlichen 5 Fällen eine Minderung. Eine detailliertere Untersuchung dieses Sachverhalts folgt im Abschnitt 4.4.

Neben den durchschnittlichen F-Maßen über alle Datensets ist es von Interesse, die einzelnen Werte pro Aktie bzw. pro Aktiengruppe zu analysieren. Im Anhang A.4 sind die F-Maße pro Aktie für jeden Klassifikator als Säulendiagramm zu sehen. Eine horizontale Linie auf der Höhe des F-Maßes von 0,500 hilft dem Leser bei der graphischen Auswertung. Es ist zu sehen, dass sich die Veränderung im F-Maß mit steigendem Zeithorizont zwischen den einzelnen Aktien systematisch – für alle fünf Klassifikatoren – unterscheidet. Die Unterschiede zwischen den Aktien, gemessen in Prozentpunkten, nehmen mit steigendem Horizont signifikant zu. Auffällig ist, dass verschiedene Aktien ihr höchstes F-Maß in unterschiedlichen Horizonten erreichen. So sind die Klassifikatoren für XRX tendenziell auf den Horizonten von 1d bis 20d besser, für AMZN jedoch auf 65d und 250d. Es ist also offensichtlich, dass sich die Lernfähigkeit der Klassifikatoren pro Aktie unterscheidet. Die Idee ist nun, die Aktien anhand ihrer Kursverläufe zu gruppieren und dann genauer zu untersuchen, wie erfolgreich die Klassifikation pro Gruppe ist. Mit dieser Frage beschäftigt sich Unterabschnitt 4.1.6. Eine tabellarische Darstellung von Recall, Precision und F-Maß für jede Aktie findet sich in Anhang A.5. Aus Platzgründen sind die einzelnen, einigen Hunderte Konfusionsmatrizen nicht angehängt.

4.1.2 Auswertung auf Aktien-Ebene

Die F-Maß-Durchschnitte über alle Datensets haben erste Erkenntnisse über die verschiedenen Klassifikatoren ergeben. Die F-Maße pro Aktie allerdings, die in diese Durchschnitte einfließen, unterscheiden sich teilweise signifikant. Deshalb sollten diese separat untersucht werden. Zu diesem Zweck sind in Abbildung 4.2 die F-Maße der Klassifikatoren pro Aktie dargestellt. Es handelt sich dabei um die durchschnittlichen F-Maße der vier Klassifikatoren, die in Unterabschnitt 4.1.1 als sinnvoll einsetzbar befunden wurden: Decision Tree und Random Forest, jeweils mit und ohne Tuning.

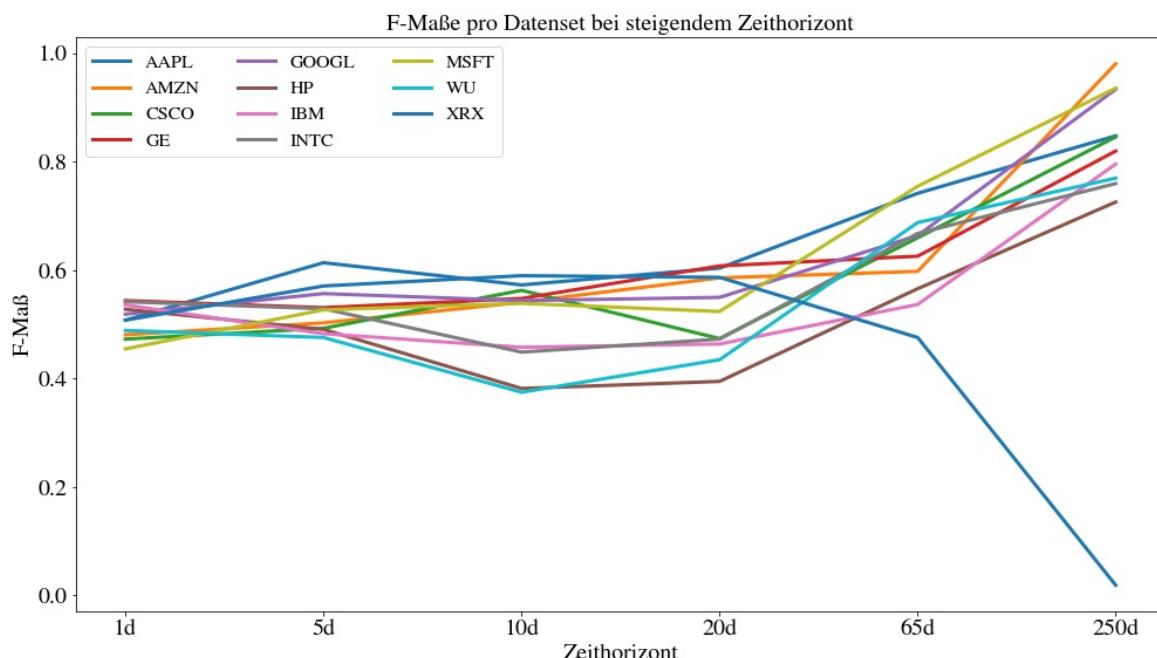


Abbildung 4.2 F-Maße der vier Klassifikatoren (DT, RF, TunedDT, TunedRF) pro Aktie

Die granularisierten Ergebnisse lassen erkennen, dass die F-Maße auf den einzelnen Aktien – ähnlich zu den Durchschnittswerten über alle Aktien – tendenziell mit einem längeren Zeithorizont steigen. Während zehn Aktien diesen Trend verfolgen, sticht XRX als Ausnahme heraus.

Tabelle 4.2 Konfusionsmatrix der vier Klassifikatoren für den 250d-Horizont auf XRX

		\hat{y} (Ergebnis der Klassifikation)	
		1	0
y	1	8 (Wahre Positive)	148 (Falsche Negative)
	0	669 (Falsche Positive)	1.895 (Wahre Negative)

Die XRX-Kurve nimmt ab dem 20d-Horizont kontinuierlich ab und erreicht bei 250d das minimale F-Maß von unter 2%. Um diesen Wert zu verstehen, ist die Konfusionsmatrix in

Tabelle 4.2 geeignet, die sich aus den Vorhersagen der vier Klassifikatoren zusammensetzt. Jedes Modell ist mit 680 Klassifikationen eingeflossen, sodass insgesamt 2.720 Werte zur Verfügung stehen. Man erkennt, dass deutlich mehr als die Hälfte der Instanzen korrekt klassifiziert wurden. Denn die Anzahl an wahren Negativen (1.895) ist größer als die der anderen drei Felder kumuliert (825). Zudem ist auffällig, dass der Großteil der korrekten Vorhersagen auf negativen Instanzen erfolgt ist: 1.895 wahre Negative stehen 8 wahren Positiven gegenüber. Aus den Konfusionswerten ergeben sich $Precision_{XRX}$, $Recall_{XRX}$ und $F - \text{Maß}_{XRX}$ wie folgt:

$$Precision_{XRX} = \frac{WP}{WP + FP} = \frac{8}{8 + 669} = 1,2\%.$$

$$Recall_{XRX} = \frac{WP}{WP + FN} = \frac{8}{8 + 148} = 5,1\%.$$

$$F - \text{Maß}_{XRX} = 2 \times \frac{Precision_{XRX} \times Recall_{XRX}}{Precision_{XRX} + Recall_{XRX}} = 1,9\%.$$

Diese Werte können aus dem Anhang A.5 hergeleitet werden. Dazu bildet man jeweils die Durchschnitte von Precision, Recall und F-Maß auf dem XRX-Datenset auf dem 250d-Horizont über die vier Klassifikatoren.

Im Vergleich zu den XRX-Resultaten zeigt die Konfusionsmatrix in Tabelle 4.3 ein Beispiel für eines der anderen zehn Datensets. Auf dem 250d-Horizont von GE ergeben sich ein Recall von 86,8%, eine Precision von 77,5% und ein F-Maß von 81,9%.

Tabelle 4.3 Konfusionsmatrix der vier Klassifikatoren für den 250d-Horizont auf GE

		\hat{y} (Ergebnis der Klassifikation)	
		1	0
y	1	1.337 (Wahre Positive)	203 (Falsche Negative)
	0	388 (Falsche Positive)	792 (Wahre Negative)

Diese Ergebnisse demonstrieren einen zentralen Nachteil des F-Maßes als Gütemaß. Aufgrund der Natur des F-Maßes wird der Erfolg eines Klassifikators nur auf den positiven Instanzen bewertet, und nicht auf den negativen Instanzen. In diesem Fall liegen die Modelle bei 73,9% der negativen Instanzen und bei 5,1% der positiven Instanzen richtig. Die niedrige Trefferrate auf den positiven Instanzen führt zu dem sehr niedrigen F-Maß von 1,9%. Die wahren Negativen bleiben bei der F-Maß-Berechnung unberücksichtigt.

Betrachtet man nun anstelle des F-Maßes zum Beispiel die Trefferrate T_{XRX} , so ergibt sich:

$$T_{XRX} = 1 - E_{XRX} = 1 - \frac{|FP| + |FN|}{N} = 1 - \frac{669 + 148}{2.720} = 70,0\%.$$

Die Trefferrate befürwortet also ein positiveres Urteil über die Klassifikatoren als das F-Maß. Mit 70% liegt sie deutlich über dem F-Maß von 1,9% und deutlich über der 50%-Marke. Mit dieser Trefferrate für den 250d-Horizont würde die XRX-Kurve einen ähnlichen Verlauf wie die anderen Datensets vorweisen. Es lässt sich festhalten, dass die Diskrepanz zwischen

F-Maß und Trefferrate auf dem XRX-Datenset signifikant ist. Auch für die anderen zehn Datensets ist die Eigenschaft des F-Maßes, dass sie die Genauigkeit nur auf der positiven Klasse erfasst, zu beachten.

Abschließend ist anzumerken, dass sich das F-Maß nur dann als Gütemaß eignet, wenn der Anwender ausschließlich auf steigende Kurse setzt und den Erfolg des Klassifikators deshalb nur auf den positiven Instanzen bewerten möchte. Allgemeiner ist das F-Maß geeignet, wenn der Erfolg nur auf einer Klasse gemessen werden soll. Powers (2014) setzt sich genauer mit dem F-Maß auseinander und zeigt gezielt Schwächen auf. Andernfalls, wenn für den Investor auch fallende Kurse relevant sind, bieten sich andere Metriken, wie die Trefferrate, an, um den Erfolg des Klassifikators auf beiden bzw. auf allen Klassen zu bewerten.

4.1.3 Veränderung der Klassen-Häufigkeitsverteilungen bei variablem Zeithorizont

Nach der Erkenntnis aus Unterabschnitt 4.1.2 ist auf eine Beobachtung hinzuweisen, die mit der gefundenen F-Maß-Schwäche zusammenhängt. Die Abbildung 4.3 zeigt, wie sich die Abweichungen der Klassen-Häufigkeiten vom Gleichgewicht entwickeln.

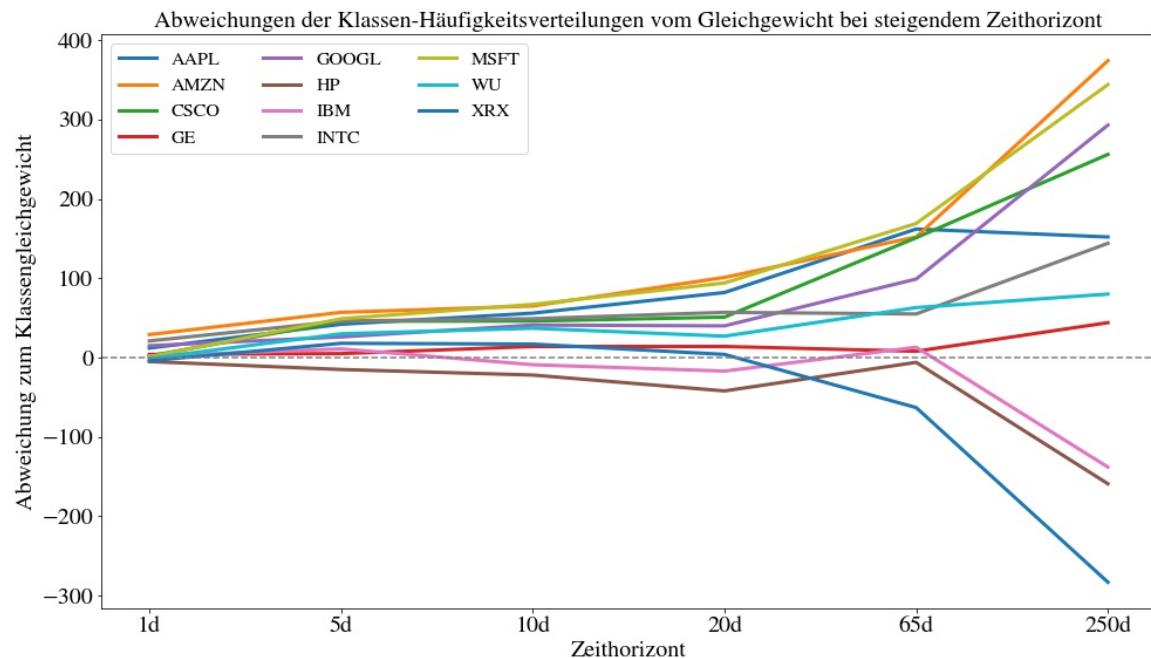


Abbildung 4.3 Zunahme der Ungleichheit der Klassen-Häufigkeiten pro Horizont

Das Gleichgewicht meint hier eine 50%-50% Verteilung zwischen der positiven und der negativen Klasse, und ist durch eine gestrichelte graue Linie bei $y=0$ dargestellt. Bei 759 Instanzen liegt dann ein Gleichgewicht vor, wenn $759 * 50\% = 380$ der Instanzen positiv sind und 379 der Instanzen negativ sind. Die Y-Achse zeigt die Einseitigkeit in Richtung der positiven bzw. der negativen Klasse an. In der Graphik bedeutet ein positiver Y-Wert y_+ , dass die positive Klasse y_+ mal öfter vorkommt als es im Gleichgewicht der Fall wäre, also insgesamt $380 + y_+$ mal. Dementsprechend kommt dann die negative Klasse $379 - y_+$ mal vor. Andererseits bedeutet ein negativer Y-Wert y_- , dass die negative Klasse insgesamt

$379 + y_-$ mal auftritt. Im Anhang A.3 sind die Klassen-Häufigkeitsverteilungen aller Datensets für jeden Horizont gezeigt. Die horizontale Linie zeigt die 50%-Linie an, die über und unter der Hälfte der Instanzen verläuft.

Es ist zu sehen, dass die Verteilung zwischen positiven und negativen Instanzen für die kurzfristigen Horizonte relativ ausgeglichen ist, dann jedoch immer mehr vom Gleichgewicht – nach oben oder nach unten – abweicht. Je länger der Horizont, desto einseitiger ist also die Klassenverteilung. Eine Ausnahme ist GE mit nahezu gleichmäßiger Verteilung für alle Horizonte. Ungleichmäßige Klassenverteilungen bergen das Risiko, dass die auf ihnen trainierten Klassifikatoren eine Verzerrung erlernen und die überrepräsentierte Klasse bevorzugen. Die Genauigkeit auf Instanzen der unterrepräsentierten Klassen leidet darunter. Um dieses Problem zu bekämpfen, haben Forscher unterschiedliche Maßnahmen vorgeschlagen. Diese Methoden würden den Rahmen der Arbeit sprengen, werden aber an dieser Stelle kurz vorgestellt. Zwei Ansätze sind das Under- und das Oversampling (Anand u. a., 2010). Beim Undersampling werden Instanzen der überrepräsentierten Klasse aus den Trainingsdaten entfernt, um das Gleichgewicht zwischen den Klassen herzustellen. Beim Oversampling hingegen werden Instanzen aus der unterrepräsentierten Klasse zum selben Zweck vervielfältigt. Die ADASYN-Methode berücksichtigt bei der Herstellung solcher neuen Instanzen zudem die Schwierigkeitsgrade der einzelnen Beispiele, indem sie fehleranfällige Instanzen häufiger wählt (Haibo He u. a., 2008). Die genannten Ansätze können in Zukunft dabei helfen, die Klassifikatoren insbesondere auf den 250d-Horizonten besser zu trainieren und somit die Genauigkeit auf der unterrepräsentierten Klasse zu erhöhen.

4.1.4 Vergleich der Lernfähigkeiten der Modelle

Die bisherigen Ergebnisse ermöglichen einen Vergleich der Lernfähigkeiten der Modelle. Eine hohe Lernfähigkeit bedeutet dabei, nützliche Regeln aus den Beispieldaten zu extrahieren um eine möglichst genaue Klassifikation zu erreichen. Dazu werden zwei Ergebnisse verwendet: die F-Maße der Modelle pro Datenset (Anhang A.4) und die Klassen-Häufigkeitsverteilungen der Datensets (Anhang A.3). Um die Lernfähigkeit der Modelle zu untersuchen, werden zuerst Abschnitte in den Klassen-Häufigkeitsverteilungen gesucht, auf denen die Verteilung nahezu unverändert ist. Ob und wie sehr ein Klassifikator auf solchen Abschnitten steigt oder fällt, gibt Hinweise darauf, wie er lernt und ob er dazu in der Lage ist, aufgrund seiner Lernfähigkeit auch bei gleichbleibender Verteilung signifikante F-Maß-Steigerungen zu erzielen.

Betrachtet man zudem in Abbildung 4.1 die Entwicklung der F-Maße mit steigendem Horizont, so fällt auf, dass die höchsten Wachstumsraten des F-Maßes, zwischen den ausgewählten Horizonten, von dem 65d- auf den 250d-Horizont stattfindet. Deshalb soll zweitens eruiert werden, wodurch diese starke Steigerung – insbesondere zwischen den längsten Horizonten 65 und 250d – zustande kommt.

Die erste Beobachtung bezüglich des Dummy Klassifikators ist, dass er von einseitigeren Klassen-Verteilungen stark profitiert. Betrachtet man einen Aktienkurs, der eine relativ konstante Klassenverteilung aufweist, so verhärtet sich dieser Verdacht. GE hat, über die Horizonte hinweg, die gleichmäßigste Klassenverteilung. Bis einschließlich 65d liegt diese bei nahezu 50%, bei 250d knapp unter 50%. Ein Klassifikator, der nur mit einer ungleicheren Klassenverteilung höhere F-Maße erreicht, begegnet auf GE einer schwierigen Herausforderung. Der Dummy erreicht zwischen 1d und 65d ausschließlich Werte im Bereich zwischen

0,470 und 0,544. Für 250d erzielt er 0,623. Die anderen vier Klassifikatoren hingegen liegen auf langfristigen Horizonten signifikant über diesen Werten. Sie erreichen im Schnitt 0,626 auf 65d und 0,820 auf 250d.

Ein anderes Beispiel ist INTC. Dort verändert sich die Klassenverteilung von 20d auf 65d kaum. Die F-Maße auf 20d und auf 65d des Dummy Klassifikators – 0,609 und 0,602 – sind nahezu unverändert. Die F-Maße der komplexeren Modelle hingegen steigen signifikant von durchschnittlich 0,473 auf 0,668. Auch auf relativ konstanten Abschnitten auf AAPL (von 65d auf 250d), HP (von 20d auf 65d) und WU (von 65d auf 250d) ergibt sich dasselbe Fazit. Es ist allerdings zu beachten, dass dieser Effekt verstärkt auf den längerfristigen Horizonten ab 20d auftritt. Vor 20d sind die Änderungen der Zeitspannen wesentlich geringer. Hier lassen sich keine klaren Muster erkennen.

Während also die Klassenverteilung über die Horizonte nahezu gleich bleibt, sind die komplexeren Modelle dazu in der Lage, für längere Zeithorizonte signifikant höhere F-Maße zu erzielen als für kürzere. Daraus folgt, dass sie eine höhere Lernfähigkeit als der Dummy besitzen, für den sich eine solche F-Maß-Zunahme nicht beobachten lässt. Der Grund für den Erfolg der Baum-basierten Modelle könnte in deren Gewichtung der Features liegen. Dank dieser Gewichtung sind die Modelle flexibler in ihrem Lernprozess und können die aussagekräftigsten Features erkennen. Dies ermöglicht die beobachteten signifikanten F-Maß-Steigerungen bei gleichbleibenden Klassenverteilungen. Ein detaillierter Vergleich diesbezüglich, zwischen dem Decision Tree und dem Random Forest, folgt in Abschnitt 4.2.

4.1.5 Precision und Recall

Neben dem F-Maß zeigt die Tabelle 4.1 auch Recall- und Precision-Werte. Es ist erkennbar, dass der Dummy Klassifikator in allen bis auf den letzten Horizont einen höheren Recall- als Precision-Wert erzielt. Das heißt, der Klassifikator erkennt zwar viele der positiven Instanzen korrekterweise als positiv. Wenn er allerdings Instanzen als positiv bestimmt, dann hat er dort eine vergleichsweise niedrigere Erfolgsquote, richtig zu liegen. Er neigt also dazu, eher falsche Positive zu deklarieren als positive Instanzen als negativ zu verpassen, als die anderen Modelle. Denn die vier Decision Tree und Random Forest Modelle weisen in 19 von insgesamt 24 Fällen (4 Klassifikatoren mit jeweils 6 Horizonten) eine höhere Precision als Recall auf. Somit neigen diese wiederum – verglichen mit dem Dummy Klassifikator – eher dazu, falsche Negative zu bestimmen. Das heißt, diese vier Modelle verpassen zwar eher eine Investitionsmöglichkeit, als der Dummy. Wenn sie aber eine Investitionsmöglichkeit, also einen steigenden Kurs, identifizieren, dann ist diese mit einer höheren Wahrscheinlichkeit auch korrekt. Je nach Anwendungsfall eignet sich also der eine oder der andere Klassifikator besser.

Ob im betrachteten Anwendungsfall der Recall- oder der Precision-Wert wichtiger ist und maximiert werden sollte, hängt stets vom Investor ab, der den Klassifikator einsetzt. Hier wird die Annahme getroffen, dass die Anlageentscheidungen eines Investors möglichst hohe Renditen bei minimalem Risiko einbringen sollen. Um die Präferenz eines Investors zu finden, werden zwei Extremfälle verglichen. Im ersten Szenario liegen 100 Kaufempfehlungen für Aktien vor, von denen 80 Stück im betrachteten Horizont tatsächlich steigen. Der Anleger hat also pro Aktie eine Erfolgswahrscheinlichkeit von 80%. Im zweiten Szenario liegen nur 20 Kaufempfehlungen vor, von denen aber 19 Stück im betrachteten Horizont

tatsächlich steigen. Dann liegt die Erfolgswahrscheinlichkeit pro Aktie bei 95%. Ohne weitere Annahmen über die Risikobereitschaft zu treffen, lässt sich keine eindeutige Präferenz des Investors bestimmen. Möchte der Investor um jeden Preis Verluste vermeiden und ist dafür bereit, möglicherweise Gewinne zu verpassen, so würde er Szenario 2 vorziehen. Möchte er hingegen eine möglichst hohe Anzahl an steigenden Aktien besitzen und nimmt dafür mehr Verlustinvestitionen in Kauf, so würde er Szenario 1 vorziehen. Zur Anwendung der Klassifikatoren in der Realität muss also die Risikobereitschaft des Anlegers bekannt sein, um die richtigen – bezogen auf die Anleger-spezifische Optimierung des Recall- bzw. Precision-Wertes – Modelle einzusetzen.

4.1.6 Auswertung pro Aktiengruppe

Es folgt eine Analyse der beobachteten F-Maß-Differenzen der Klassifikatoren zwischen den einzelnen Aktiengruppen. Beim Vergleich der F-Maße pro Aktie im Anhang A.4 fällt auf, dass XRX bei 250 Tagen für jeden Klassifikator unter 0,100 liegt, und somit dem, gemessen am Durchschnitt, allgemeinen Trend – einem steigenden F-Maß bei steigendem Horizont – widerspricht. Man betrachte nun den Aktienverlauf von XRX im Anhang A.1. Bis 2015 steigt die Aktie in einem deutlich positiven Trend, anschließend jedoch fällt sie kontinuierlich bis 2017. Von 2017 bis zum letzten Datensatz aus 2018 lässt sich der Kurs als alternierend beschreiben.

Gleichzeitig liegen die F-Maße aller Klassifikatoren für AMZN mit Zeithorizont 250d bei über 96%. Der Aktienkurs von AMZN verfolgt über den gesamten Beobachtungszeitraum von fünf Jahren einen Aufwärtstrend, mit nur wenigen kurzweiligen Ausnahmen. Es kommt die Vermutung auf, dass der Kursverlauf der Aktie Einfluss auf die F-Maß Entwicklung der Klassifikatoren hat. Deshalb werden die elf Datensets für eine genauere Analyse in drei Gruppen unterteilt. Das Kriterium für die Zuordnung einer Aktie in eine Gruppe ist der Trend ihres Kursverlaufs. Die Einteilung ist in Tabelle 4.4 dargestellt.

Tabelle 4.4 Einteilung der Aktien in Gruppen zur weiteren Analyse

Gruppe	Kursverlauf	Datensets
1	Steigend	AAPL, AMZN, CSCO, GOOGL, INTC, MSFT
2	Alternierend	GE, HP, WU, XRX
3	Fallend	IBM

Die Liniendiagramme 4.4, 4.5 und 4.6 zeigen die relativen Kursentwicklungen der Aktien pro Gruppe ausgehend vom 08.02.2013. Stellt man sich den Zeithorizont der Klassifikation als horizontale Linie vor, so kann man Vermutungen anstellen, wie sich eine Verlängerung dieses Horizonts bzw. der gedachten horizontalen Linie auf den Klassifikator auswirkt. In Abbildung 4.4 kann man so feststellen, dass auf den steigenden Aktien ein längerer Horizont in den meisten Fällen zu einer deutlicheren Entscheidung führt.

Sucht man sich in Abbildung 4.4 einen der sechs Aktienkurse aus und stellt sich einen 30d-Horizont vor, den man von 2013 startend horizontal an den Tagespunkt anhängt, so finden sich zwar sehr viele positive Instanzen, bei denen der Wert am Ende des Horizonts größer als der aktuelle Tageswert ist. Aber es gibt auch negative Instanzen, bei denen der

Kurs in den kommenden 30 Tagen sinkt. Verlängert man nun den Horizont auf 5 Jahre, also auf die komplette vorhandene Zeitdauer, so kommt man bei jeder Klassifikation zu einem positiven Ergebnis. Der Klassifikator könnte also von möglichst langen Horizonten profitieren, da die Entscheidungen eindeutiger werden.

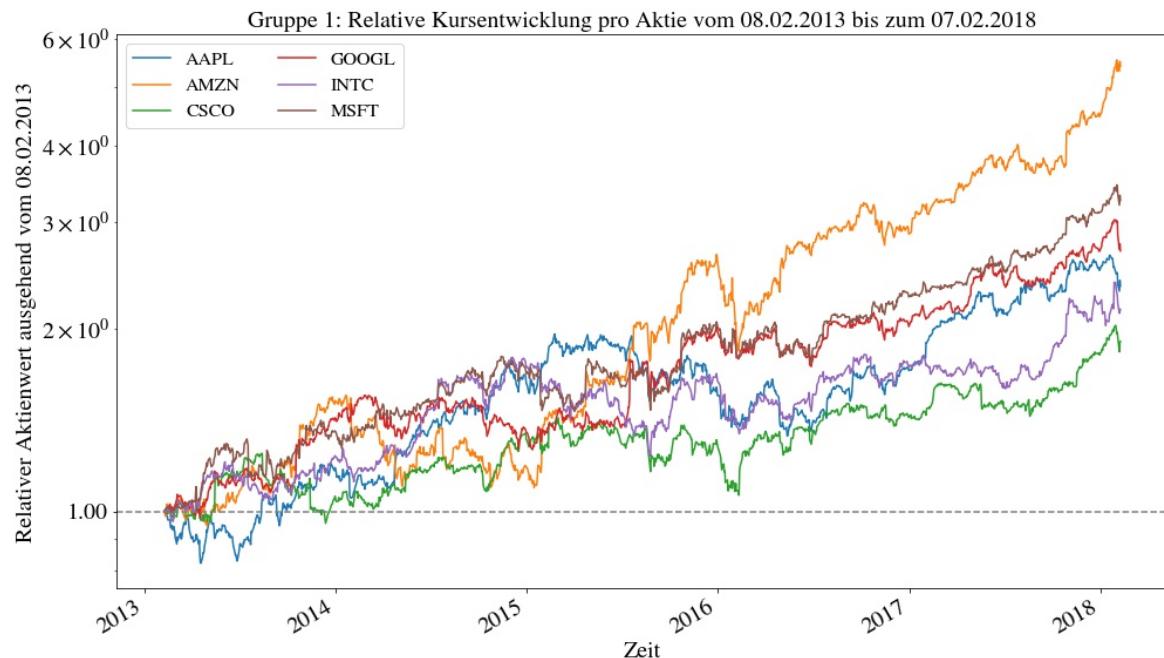


Abbildung 4.4 Relative Kursentwicklung der Gruppe 1-Aktien (steigend)

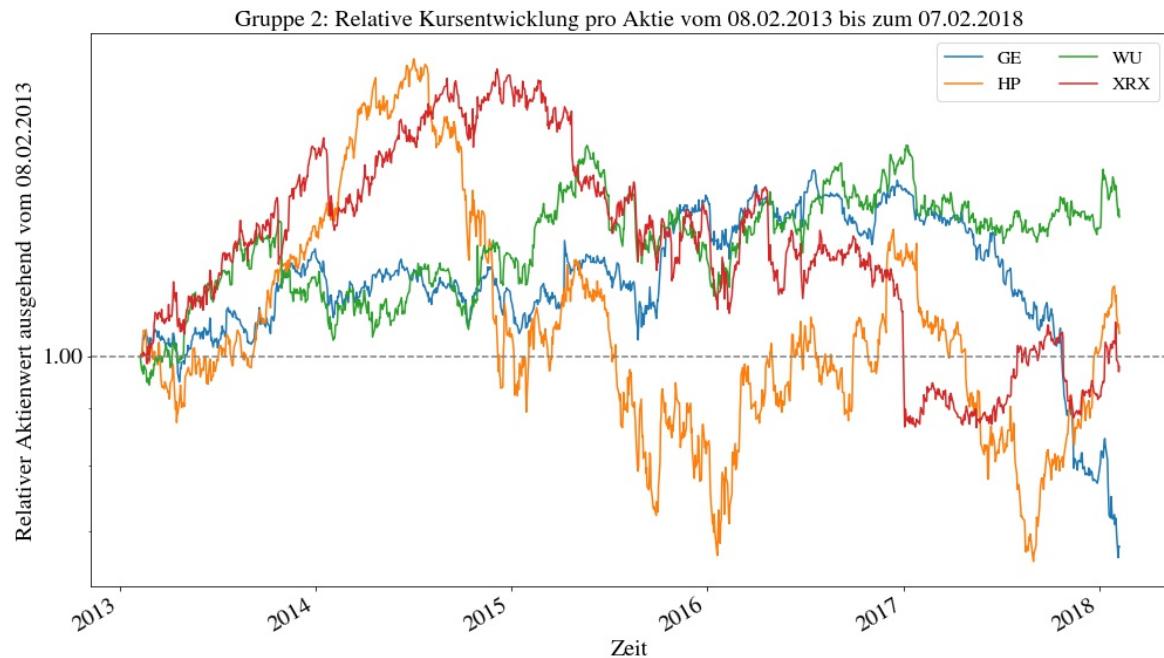


Abbildung 4.5 Relative Kursentwicklung der Gruppe 2-Aktien (alternierend)

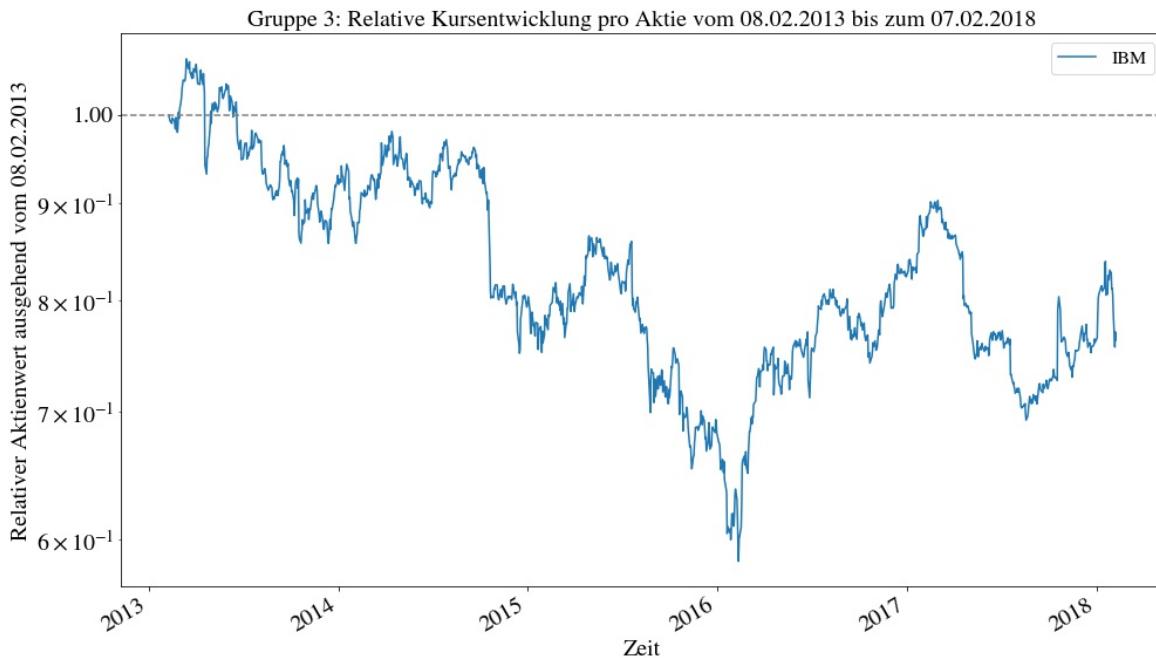


Abbildung 4.6 Relative Kursentwicklung der Gruppe 3-Aktie (fallend)

Ähnlich verhält es sich in Abbildung 4.6, nur in die andere Richtung: je länger der Horizont, desto mehr negative Instanzen sind zu erkennen. Für die Aktien aus Gruppe 2 gelten die Regel allerdings nicht. Abhängig von den Wellenlängen können verschiedene Zeithorizonte zum höchsten F-Maß führen, es lässt sich keine simple Regel formulieren.

Um die Klassifikation pro Horizont besser zu verstehen, ist zu beachten, dass – unabhängig vom Zeithorizont – die Klassifikatoren im Training jeweils Features über verschieden lange historische Zeiträume erhalten. Diese Features bilden Zeithorizonte in die Vergangenheit ab. Die Zeitspannen sind, wie bereits in Abschnitt 3 aufgeführt, dieselben wie bei den vorwärts-gerichteten Horizonten. Der Verdacht liegt nahe, dass für längere Anlagehorizonte die längerfristigen Features besser geeignet sind als kurzfristige, und vice versa. In dem Fall sollte der Klassifikator erkennen, welche Features über welchen vergangenen Zeitraum für seine Entscheidung am wichtigsten sind, und diese entsprechend hoch gewichten. Um das zu untersuchen, werden die Einflussgrade der einzelnen Features in Abhängigkeit vom Zeithorizont analysiert. Diese Auswertung findet in Abschnitt 4.2 statt.

Wie sich die Klassifikatoren auf den drei Gruppen verhalten, ist in Abbildung 4.7 zu sehen. Das Diagramm zeigt die F-Maße pro Gruppe in Abhängigkeit vom Zeithorizont. Jedes F-Maß ist als Durchschnitt über die vier Klassifikatoren, die in Unterabschnitt 4.1.1 als sinnvoll einsetzbar befunden wurden, errechnet. Wieder zeigt die X-Achse die Zeithorizonte in verzerrter Skalierung, sodass die äquidistanten Markierungen zunehmende Zeitintervalle bedeuten.

Es lässt sich festhalten, dass die Klassifikatoren auf den Aktien mit Aufwärts- und Abwärts-trends (Gruppe 1 und Gruppe 3) ab dem 20-Tage-Horizont mit längerem Horizont bessere Ergebnisse erzielen. Bei 250d erreichen beide Gruppen ein deutliches Maximum. Für die Horizonte vor 20d weist Gruppe 1 ein schwaches Wachstum auf. Gruppe 3 erfährt zwischen

1d und 10d eine negative Entwicklung. Die Verzerrung der X-Achse führt anschließend zu der stark erhöhten Steigung der Kurve, vor allem zwischen 65d und 250d. Auf einer linearen X-Achse sieht die Steigung der Kurve entsprechend geringer aus. Auf den alternierenden Daten hingegen, aus der Gruppe 2, stagniert das F-Maß ab 65d. Die Klassifikatoren profitieren hier nicht von den zusätzlichen 185 Tagen, die mit dem 250d-Horizont hinzukommen. Das ist mit der vorherigen Beobachtung, dass sich für Gruppe 2 graphisch gesehen eine Horizontverlängerung nicht immer positiv auswirken muss, konsistent.

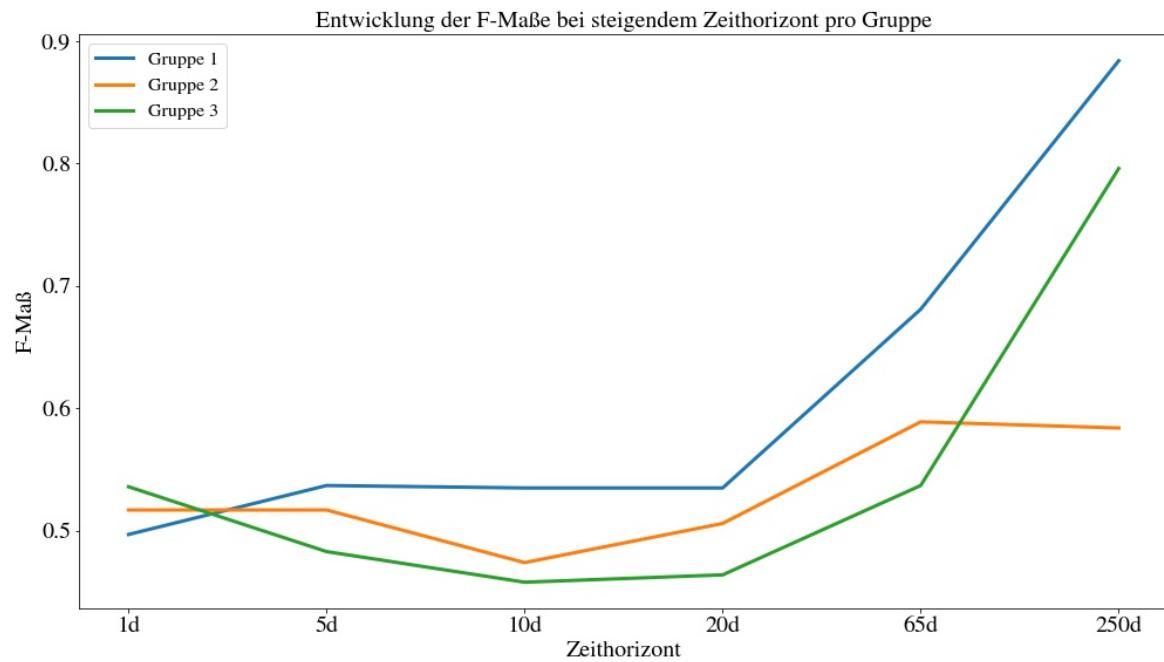


Abbildung 4.7 Durchschnittliche F-Maße der vier Baum-basierten Klassifikatoren pro Gruppe

Es ist zu beachten, dass in diesen Untersuchungen die Gruppe 2 aus vier ausgewählten alternierenden Aktien besteht, wodurch eine Verzerrung gegenüber der Gesamtheit alternierender Aktien entsteht. Dies gilt auch für Gruppe 1 und insbesondere für Gruppe 3. Für andere alternierende Aktien kann der Höhepunkt im F-Maß an einem anderen Zeithorizont liegen. Auch ist es denkbar, dass bei wellenförmigem Kursverlauf mehrere Zeithorizonte zu gleich- oder ähnlich-hohen lokalen Maxima führen. Beträgt die Wellenlänge den Wert l und liegt ein lokales Maximum bei p vor, dann könnten alle Punkte aus $\{q \mid q \in p + l \times n, n \in \mathbb{Z}\}$ ebenfalls lokale Maxima darstellen. Wie viele dieser Punkte tatsächlich zu lokalen Maxima führen, hängt davon ab, wie viele Perioden lang sich die Welle im Aktienkurs wiederholt und wie variabel die Wellenlänge ist. Vorhersagen diesbezüglich könnten dann möglich sein, wenn beispielsweise jährliche Vorkommnisse, wie die Veröffentlichung des Jahresberichts oder Ankündigungen von Zentralbanken, einen absehbaren Effekt auf den Aktienkurs haben. Um Klassifikatoren für solche Prognosen zu trainieren, sollten Unternehmen verschiedener Branchen verglichen werden. Es ist zum Beispiel vorstellbar, dass Aktienkurse von Unternehmen mit überdurchschnittlich hohem Finanzierungsbedarf stärker auf Leitzinsänderungen reagieren. Die Vorhersage, wie ein solches wiederkehrendes Vorkommnis den Aktienkurs beeinflusst, ist von vielen externen Faktoren abhängig. So ist es möglich, dass eine Entscheidung der europäischen Zentralbank von vorangegangenen Entscheidungen anderer

Notenbanken abhängt. Kann man einflussreiche Vorkommnisse und deren Einflüsse auf ausgewählte Aktienkurse prognostizieren, so könnte man für alternierende Aktien eine optimierte Handelsstrategie entwerfen, um von jeder erkannten Welle im Kursverlauf zu profitieren.

Trotz teilweiser Einschränkungen lassen sich zwei Erkenntnisse fassen. Die vier Klassifikatoren profitieren auf alternierenden Datensets nicht unbedingt von längeren Zeithorizonten, sondern sie erreichen ein Maximum und stagnieren anschließend. Ein längerer Zeithorizont hat auf diesen Daten einen anderen Effekt, als auf Daten mit einem klaren Aufwärts- oder Abwärtstrend. Denn für letztere Aktien lässt sich eine signifikante F-Maß Steigerung für längere Horizonte feststellen.

Die ausführlichen Ergebnisse jedes Klassifikators pro Aktiengruppe finden sich im Anhang A.8. Es fällt auf, dass alle Decision Tree und Random Forest Varianten auch in Gruppe 3 – also auf Aktien, die einen Abwärtstrend verfolgen – in den Horizonten 65d und 250d eine deutliche Steigerung des F-Maßes erreichen. Die F-Maß Maxima dieser Modelle bei 250d liegen bei circa 80%. Der Dummy Klassifikator jedoch weist in den zwei längsten Horizonten fallende F-Maße auf. Während er auf 1d, 5d und 10d eine Trefferrate von circa 50% erreicht, liegt diese für 250d bei knapp 30%, dem Minimum. Diese Beobachtung lässt darauf schließen, dass die komplexeren Modelle besser dazu in der Lage sind, anhand der Trainingsdaten zu lernen. Denn alle Klassifikatoren erhalten in den Trainingsepochen dieselben Trainings- und Testdatensätze. Weil der Dummy Klassifikator jedoch nur die vergangene Klassenverteilung beachtet und alleine anhand der Klassenwahrscheinlichkeiten entscheidet, führen Veränderungen in dieser Klassenverteilung zu schlechteren Prognosen. Ab dem 20d Horizont führen solche Veränderungen also zu signifikant schlechteren F-Maßen. Die Baum-basierten Klassifikatoren jedoch bewerten in jeder Trainingsepoke die Beispieldaten neu, und gewichten auch die Features von neuem. Die Ergebnisse für Gruppe 3 unterstützen somit die Aussage, dass die Baum-basierten Modelle besser als der Dummy zur Aktientrend-Klassifikation geeignet sind, da sie mit schwankenden Klassenverteilungen besser umgehen und somit flexibler sind.

4.2 Einflussgrade der Features in Abhängigkeit vom Zeithorizont

Die untersuchten Baum-basierten Modelle lernen anhand historischer Beispieldaten. Diese Daten werden durch verschiedene Features repräsentiert, wie in Kapitel 3 geschildert. Die Features werden anschließend in jedem Trainingsvorgang unterschiedlich gewichtet. Es besteht nun die Vermutung, dass ein Klassifikator abhängig vom Zeithorizont der Klasse, die er vorhersagt, jene Features höher gewichtet, welche einen entsprechend langen historischen Zeitraum erfassen. Zum Beispiel könnten zur Vorhersage für den 250d-Horizont die 250d-Features höher gewichtet werden, während zur Vorhersage für den 10d-Horizont die 10d-Features wichtiger sind. Die Annahme dabei ist, dass langfristige Prognosen vom Erkennen langfristiger Trends der Vergangenheit mehr profitieren als vom Erkennen kurzfristiger Trends, und vice versa.

Um zu untersuchen, inwiefern die Baum-basierten Modelle eine solche Gewichtung der Features vornehmen, zeigen die Abbildungen 4.8 und 4.9 die Einflussgrade aller Features bei steigendem Horizont jeweils für den Decision Tree und den Random Forest Klassifikator. Zu besseren Sichtbarkeit wurden die Features basierend auf dem Zeitraum, den sie

erfassen, in Gruppen eingeteilt und gefärbt. Die kurzfristigen (1d, 5d) Features sind in Gelbtönen zu sehen, die mittelfristigen (10d, 20d) in Grüntönen, die langfristigen (65d, 250d) in Rottönen und die Tagesfeatures (0d) in grau. Zusätzlich zeigt die rote bzw. gelbe gestrichelte Linie den Trend der Maxima der lang- bzw. kurzfristigen Feature-Gewichtungen an.

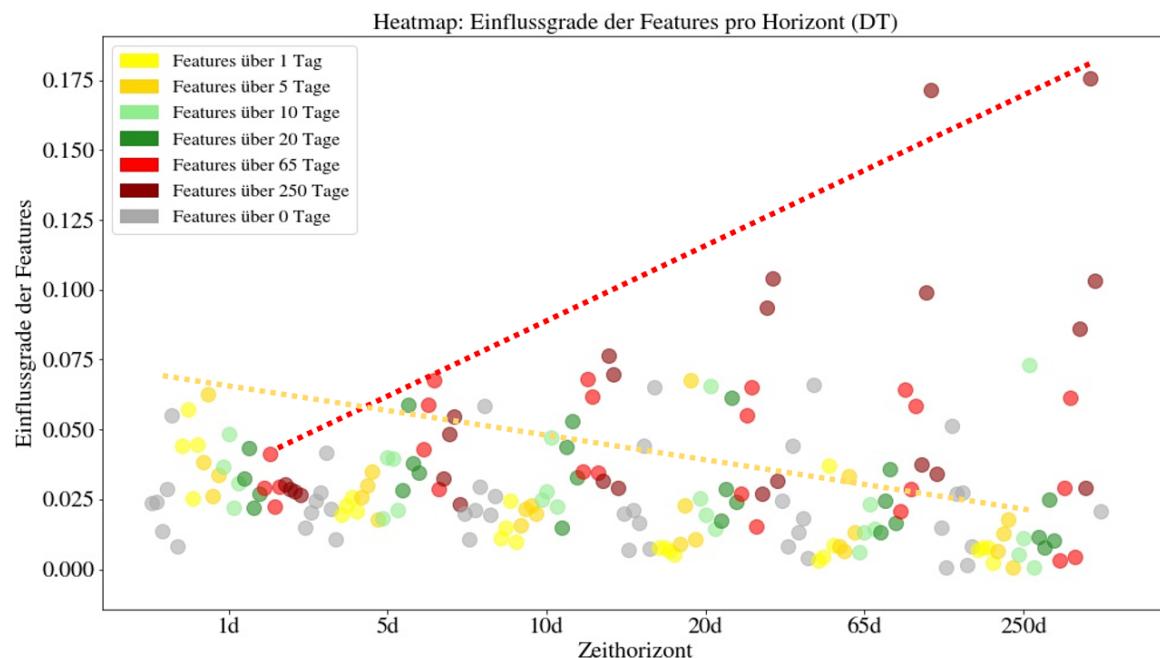


Abbildung 4.8 Einflussgrade der Features im Decision Tree Klassifikator pro Horizont

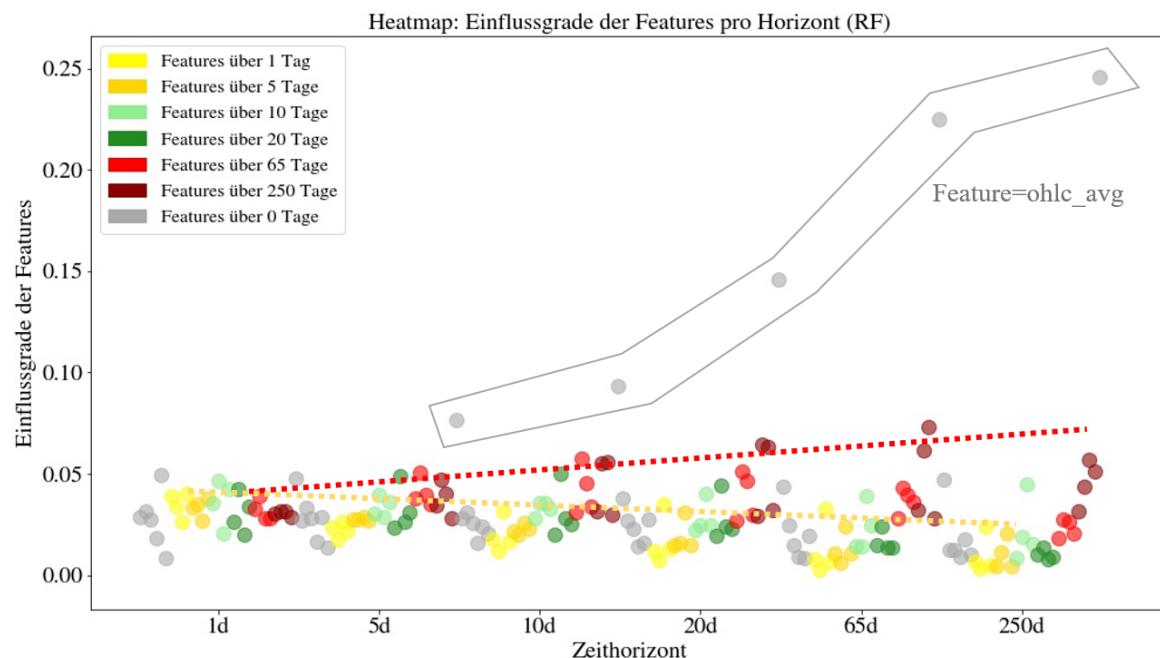


Abbildung 4.9 Einflussgrade der Features im Random Forest Klassifikator pro Horizont

Wie in den Abbildungen 4.8 und 4.9 zu sehen ist, weisen die roten Trendlinien positive Steigungen auf, und die gelben negative. Also gewichten sowohl der Decision Tree als auch der Random Forest zur Vorhersage von langen Zeithorizonten die langfristigen Features stärker, und vice versa. Zum selben Ergebnis kommt man bei Betrachtung der rot- bzw. gelbgefärbten Punktwolken, die mit steigendem Horizont ihre Mittelpunkte nach oben bzw. nach unten verlagern. Die Summen dieser Punktwolken sind in Tabelle 4.5 ersichtlich. Die Werte in der Tabelle unterstreichen die getroffenen Aussagen bezüglich der Anpassungsfähigkeit der Modelle.

Tabelle 4.5 Summen der 1d/5d- (gelb) bzw. 65d/250d-Features (rot) pro Horizont

	Horizont	1d	5d	10d	20d	65d	250d	Delta (250d-1d)
DT	Rot	0.236	0.357	0.407	0.419	0.515	0.492	+0.256
	Gelb	0.347	0.218	0.161	0.162	0.130	0.084	-0.264
RF	Rot	0.249	0.312	0.340	0.343	0.341	0.276	+0.026
	Gelb	0.319	0.277	0.260	0.290	0.327	0.325	+0.006

Es ist auffällig, dass der Random Forest das Feature "ohlc_avg" für die Horizonte ab 5d am stärksten gewichtet, sodass dessen Einflussgrade über der roten Trendlinie verlaufen. Dieses Feature ist signifikant höher-gewichtet als die restlichen. Für den 250d-Horizont ist der Einflussgrad von "ohlc_avg" circa 20% höher als der des zweitwichtigsten Features. Der Grund dafür liegt im Aufbau des Random Forests: Er kombiniert mehrere Decision Trees. Die einzelnen Bäume haben zwar einen klaren Aufwärtstrend der rot-gefärbten Features. Aber gleichzeitig ist in Abbildung 4.8 zu erkennen, dass sich die relative Gewichtung dieser rot-gefärbten Punkte untereinander von Horizont zu Horizont unterscheidet. Zum Beispiel ist "volatility_250d" für den 10d-Horizont mit 7,7% das wichtigste Feature, für den 250d-Horizont ist es jedoch "ma_250d" mit 17,6%. Zusätzlich unterscheiden sich die Feature-Gewichtungen unter den einzelnen Basisklassifikatoren, da diese auf verschiedenen zufälligen Teilmengen der Features und Daten trainiert werden.

Wenn man dann bei der Berechnung der Einflussgrade eines Random Forests den Durchschnitt über die Basisklassifikatoren verwendet, werden jene Features am höchsten gewichtet, die im Schnitt über alle Basisklassifikatoren das höchste Gewicht haben. Nach dieser Rechnung ergibt sich in Abbildung 4.9 also das Feature "ohlc_avg" als das wichtigste für den gesamten Random Forest.

Ein Grund für die hohe durchschnittliche Bedeutung des "ohlc_avg"-Wertes über die Basisklassifikatoren hinweg könnte sein, dass es die vier anderen 0d-Features (Eröffnungspreis, Tageshoch, Tagestief, Schlusspreis) kombiniert und somit eine höhere Aussagekraft hat als jedes dieser Features einzeln. Im 250d-Horizont gibt es eine solche Größe, die alle anderen 250d-Features zusammenfasst, nicht. In zukünftigen Untersuchungen bietet es sich an, für alle Horizonte solche zusammenfassenden Features einzuführen. Für die Kombination sollten neben dem Durchschnitt verschiedene Formeln ausprobiert werden, die besser geeignet sind. Zum Beispiel könnte man die Volatilität und den gleitenden Durchschnitt über eine Multiplikation kombinieren. Sei der gleitende Durchschnitt 250 und die Volatilität jeweils 1, 2 und 3. Benutzt man nun den Durchschnitt, um die beiden Größen zusammenzufassen, dann erhält man die Werte 125,5; 126; und 126,5. Die Veränderungen der Volatilität werden

sehr nur schwach widergespiegelt, da sie der deutlich kleinere Wert ist. Verwendet man stattdessen die Multiplikation, so ergeben sich 250; 500; und 750. Damit hat die Volatilität einen größeren Einfluss auf das Kombinations-Feature.

Sollte sich nach Einführung weiterer zusammenfassender Features herausstellen, dass die Modelle die Features stärker an den vorherzusagenden Zeithorizont anpassen als in dieser Arbeit, so kann dies zu Verbesserungen der Genauigkeit führen. Somit stellt dieser Bereich eine vielversprechende Ausgangspunkt für anknüpfende Forschung dar.

Die Erkenntnisse deuten auf eine hohe Lernfähigkeit und Flexibilität der Baum-basierten Modelle hin. Diese Modelle richten die Feature-Gewichtung flexibel an der vorherzusagenden Klasse aus. Der Dummy Klassifikator hingegen nimmt keine Feature-Gewichtungen vor. Er kann sich auf diese Weise also nicht an die angefragten Klassen anpassen, und ist somit weniger flexibel.

4.3 Einfluss von Feature Extraction

Die Grundlage für die bisherigen Ergebnisse sind unter anderem die Features aus der technischen Analyse. Diese Features wurden von den Klassifikatoren hoch gewichtet, wie sich in Abschnitt 4.2 herausgestellt hat. Welchen Einfluss diese Features auf die F-Maße der Modelle haben, ist in Abbildung 4.10 dargestellt. Die Säulen sind als durchschnittliche F-Maße über alle Datensets und Horizonte berechnet. Die Werte sind in Tabelle 4.6 zu sehen.

Die Ergebnisse zeigen, dass Feature Extraction bei den Baum-basierten Klassifikatoren zu höherer Genauigkeit, gemessen mit dem F-Maß, führt. Alle Decision Tree und Random Forest Varianten erzielen mit Feature Extraction höhere Werte als ohne. Der Dummy Classifier dagegen ist, wie erwartet, von den zusätzlichen Features unbeeinflusst.

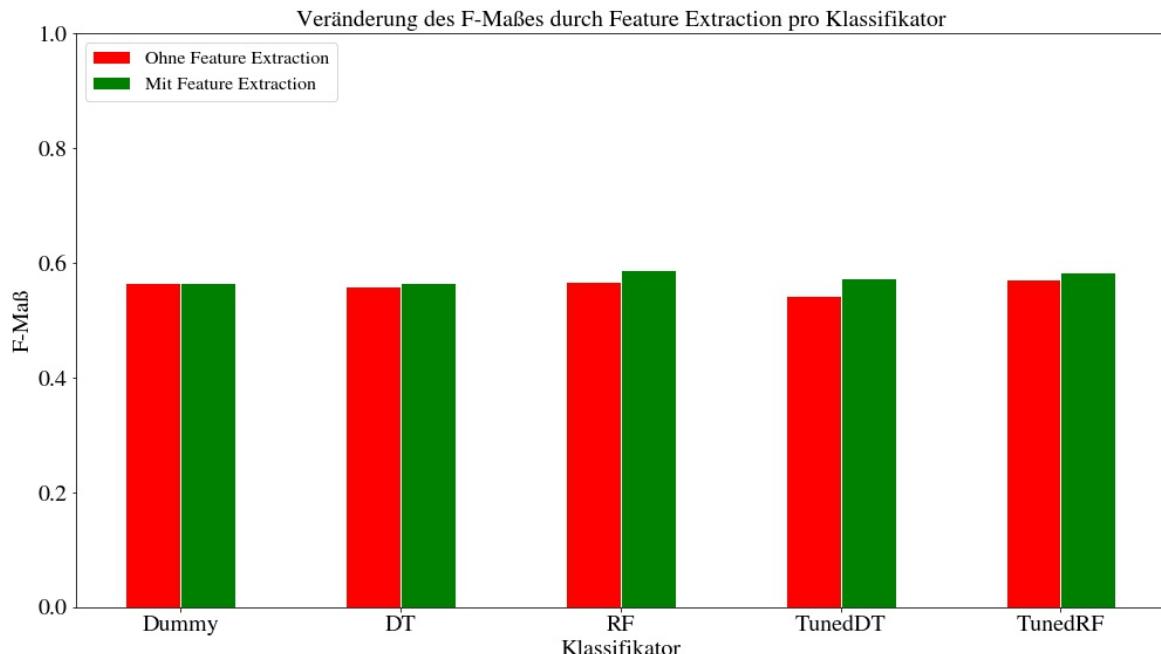


Abbildung 4.10 F-Maße der Klassifikatoren mit und ohne Feature Extraction

Tabelle 4.6 Veränderung der F-Maße pro Klassifikator durch Feature Extraction

Veränderung des F-Maßes durch Feature Extraction	Ohne Feature Extraction	Mit Feature Extraction	Differenz
Dummy	0,565	0,565	0,000
DT	0,559	0,567	0,007
RF	0,569	0,587	0,019
TunedDT	0,543	0,574	0,030
TunedRF	0,572	0,584	0,012

Um den Einfluss von Feature Extraction näher zu untersuchen, folgen nun zwei Betrachtungen der Ergebnisse auf unterschiedlichen Dimensionen. Zuerst werden die F-Maße pro Horizont untersucht. Dann erfolgt eine Aufteilung der Resultate auf die Aktiengruppen und auf die einzelnen Aktien.

In Abbildung 4.11 sind die F-Maß-Änderungen jedes Klassifikators bei steigendem Horizont als Linien veranschaulicht. Die zugrundeliegenden Werte sind in der Tabelle 4.7 enthalten.

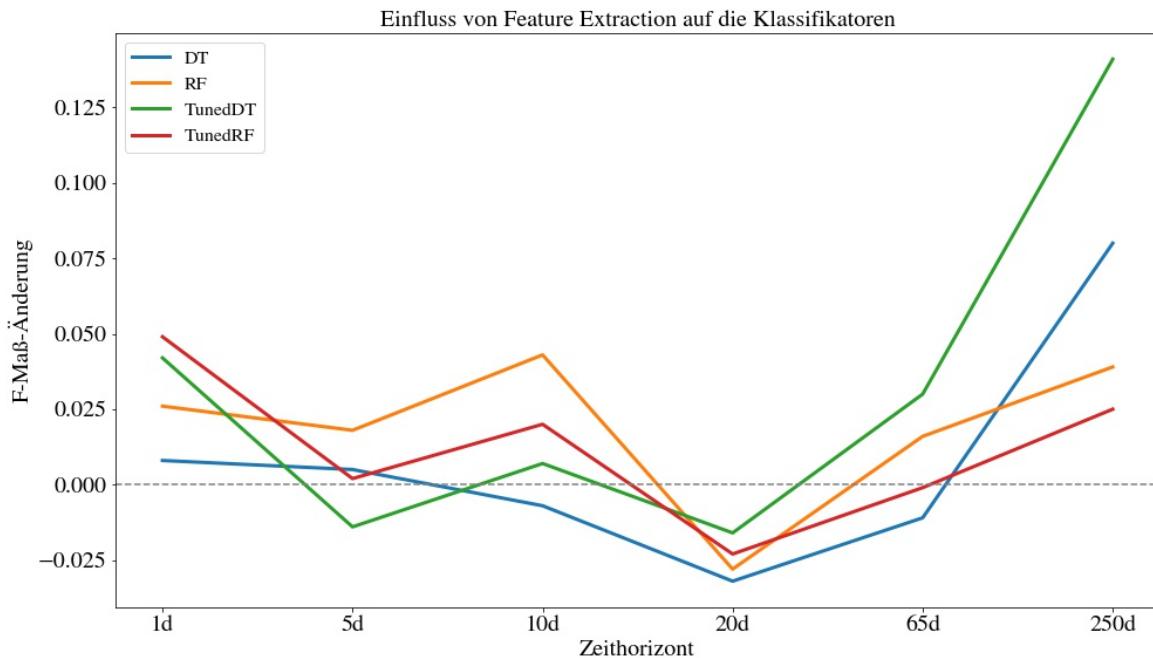


Abbildung 4.11 Einfluss von Feature Extraction auf die F-Maße bei steigendem Zeithorizont

Es lässt sich feststellen, dass die F-Maße auf sehr kurzen und sehr langen Horizonten am meisten von Feature Extraction profitieren. Für alle vier Klassifikatoren ist im 1d-Horizont ein lokales Maximum zu beobachten. Anschließend sinken die Werte tendenziell, bis sie bei 20d negative Minima erreichen. Auf den zwei längsten Horizonten steigen die F-Maße signifikant an. Die Decision Tree Modelle weisen für den 250d-Horizont deutlich höhere Werte auf als die Random Forests. So erreicht der Decision Tree ein Maximum mit einer absoluten F-Maß-Steigerung von 8%, oder mit Tuning von über 14%. Beide Random Forest

Varianten liegen im selben Horizont bei F-Maß-Zunahmen von unter 5%.

Tabelle 4.7 Veränderung der F-Maße durch Feature Extraction pro Horizont

Horizont	1d	5d	10d	20d	65d	250d
DT	0.008	0.005	-0.007	-0.032	-0.011	0.080
RF	0.026	0.018	0.043	-0.028	0.016	0.039
TunedDT	0.042	-0.014	0.007	-0.016	0.030	0.141
TunedRF	0.049	0.002	0.020	-0.023	-0.001	0.025

Der Grund für die höhere F-Maß-Steigerung der Decision Trees liegt in deren höheren Gewichtung der technischen Features. Im Abschnitt 4.2 wurde ersichtlich, dass der Random Forest die 0d-Features stärker gewichtet als der Decision Tree, insbesondere auf den langen Horizonten. Somit ist auch der Einfluss von Feature Extraction auf Random Forests schwächer als auf Decision Trees.

Die besseren Ergebnisse der Decision Trees mit Tuning, gegenüber jenen ohne Tuning, hängen mit den optimierten Parametern, wie der maximalen Tiefe des Baumes, zusammen. Angenommen, der ursprüngliche Baum (ohne Tuning) hat eine Tiefe von 10 und der optimierte Baum (mit Tuning) eine Tiefe von 3. Dann hat der ursprüngliche Baum 7 mal öfter die jeweils wichtigsten Features ausgewählt und diese zur Aufteilung der dortigen Instanzen angewendet. Das Problem dabei ist, dass die Menge der zur Verfügung stehenden Instanzen mit jeder Tiefenstufe abnimmt. Somit basiert die Entscheidung, welches Feature das wichtigste ist, in Tiefe 8 auf einem Bruchteil der Instanzen, die für dieselbe Entscheidung in Tiefe 2 gedient haben. Je weniger Instanzen in diese Entscheidung einfließen, desto weniger repräsentativ ist die Entscheidung und somit nimmt auch deren Güte ab. Das bedeutet, dass der ursprüngliche Baum also eher dazu neigt, unwichtige Features zu hoch zu gewichten, als der optimierte. Und somit profitiert dieser unkontrollierte Baum wiederum weniger von den zusätzlichen Features. Er gewichtet sie entsprechend niedriger, als es ein getrimmter Baum. Für den Random Forest lässt sich keine solche signifikante Differenz zwischen den Varianten mit Tuning und ohne Tuning feststellen.

Eine weitere Aussage lässt sich in Bezug auf den Modell-unabhängigen Nutzen der technischen Features treffen. Auf sehr kurzen und sehr langen Horizonten führen die technischen Features zu den höchsten F-Maß-Steigerungen. Mittelfristig ist ihr Einfluss teilweise negativ. Die Vermutung liegt nahe, dass die technischen Features nur langfristig signifikanten Nutzen stiften können, indem sie Trends über lange Zeiträume erkennbar machen. Mittelfristige Prognosen sind aufgrund von unvorhersehbaren Ereignissen und Reaktionen der Marktteilnehmer auch im Durchschnitt über die betrachteten Aktien nicht sinnvoll möglich, auch nicht mit Feature Extraction. Diese These wird dadurch bekräftigt, dass zwischen dem 5d-Horizont und dem 20d Horizont die Linien stark variieren. Der TunedDT Klassifikator zum Beispiel liegt auf 5d bei -1.4%, auf 10d bei positiven 7% und schließlich bei 20d bei negativen -1.6%. Weiterhin weisen die Ergebnisse aus Abschnitt 4.2 darauf hin, dass zwar kurzfristige (1d, 5d) und langfristige (65d, 250d) Features mit zunehmendem Zeithorizont entsprechend gewichtet werden. Auf die mittelfristigen (10d, 20d) Feature trifft dies jedoch nicht zu. Für diese bildet sich keine klare Trendlinie. Vereinzelt sind in Abbildung 4.8 zwischen 5d und

20d zwar höher-gewichtete grüne Punkte zu sehen, aber diese liegen stets deutlich unter den dunkelroten und roten Punkten. Die Klassifikatoren erkennen sie also selbst auf den mittelfristigen nicht als wichtig an, denn dort dominieren bereits die langfristigen Features.

In Bezug auf die vorherigen Abschnitte ist zusammenzufassen, dass Feature Extraction zwar eine positive Auswirkung auf die F-Maße der Klassifikatoren hat, diese jedoch nur bei langfristigen Horizonten signifikant ist. Die technischen Features helfen den Klassifikatoren dabei, langfristige Trends zu erkennen führen zu einer weiteren Erhöhung der ohnehin hohen F-Maßen auf den 65d- und 250d-Horizonten.

Ergänzend bestätigt auch eine Untersuchung auf Gruppen-Ebene dieses Ergebnis. Denn, wie in Tabelle 4.8 aufgeführt, sind die F-Maß-Veränderungen für Aktien aus Gruppe 1, die also einen klaren Aufwärts-Trend aufweisen, signifikant höher, als jene für alternierende Aktien aus Gruppe 2. Eine allgemeine Aussage bezüglich Gruppe 3, die eine negative F-Maß-Veränderung erfährt, lässt sich aufgrund einer einzigen zugrundeliegenden Aktie nicht treffen. Die Auswirkungen von Feature Extraction sind detaillierter auf Aktien-Ebene im Anhang A.6 zu finden.

Tabelle 4.8 Veränderung der F-Maße durch Feature Extraction pro Gruppe

Gruppe	F-Maß-Veränderung durch Feature Extraction
1	0,025
2	0,014
3	-0,013

Aktien mit einem klaren Trend profitieren also besonders stark von Feature Extraction. Das ist mit der vorherigen Erkenntnis vereinbar, dass die Klassifikatoren durch technische Features Trends besser erkennen können und somit höhere F-Maße erzielen.

4.4 Optimierung der Hyperparameter

Eine weitere Methode, mit der die Genauigkeit der Klassifikatoren verbessert werden kann, ist das Hyperparameter-Tuning. Die Güte der Hyperparameter-Kombinationen bestimmt sich nachfolgend durch das F-Maß, das über die Time-Series Cross Validation mit zehn Teilstichproben (10TSCV) ermittelt wird. Um die beiden Modelle zu vergleichen, werden die Tuning-Ergebnisse für den Decision Tree und für den Random Forest separat dargestellt. In den Abbildungen 4.12 und 4.13 sind die durch das Tuning verursachten F-Maß-Veränderungen pro Horizont, aufgeteilt nach Aktien, ersichtlich. Die letzte, schwarze Säule eines jeden Horizonts zeigt den Durchschnitt über die 11 Aktien. Die genauen Differenzen können aus den Ergebnistabellen im Anhang A.5 berechnet werden.

Wie in Abbildung 4.12 zu sehen ist, profitiert der Decision Tree auf dem 1d-Horizont von Tuning, bevor die durchschnittlichen F-Maß-Veränderungen auf 5d und 10d negativ sind. Anschließend weist er auf 20d eine minimale und auf 65d eine deutlichere Steigung auf. Auf dem 250d-Horizont ist das F-Maß nahezu unverändert (-0,001).

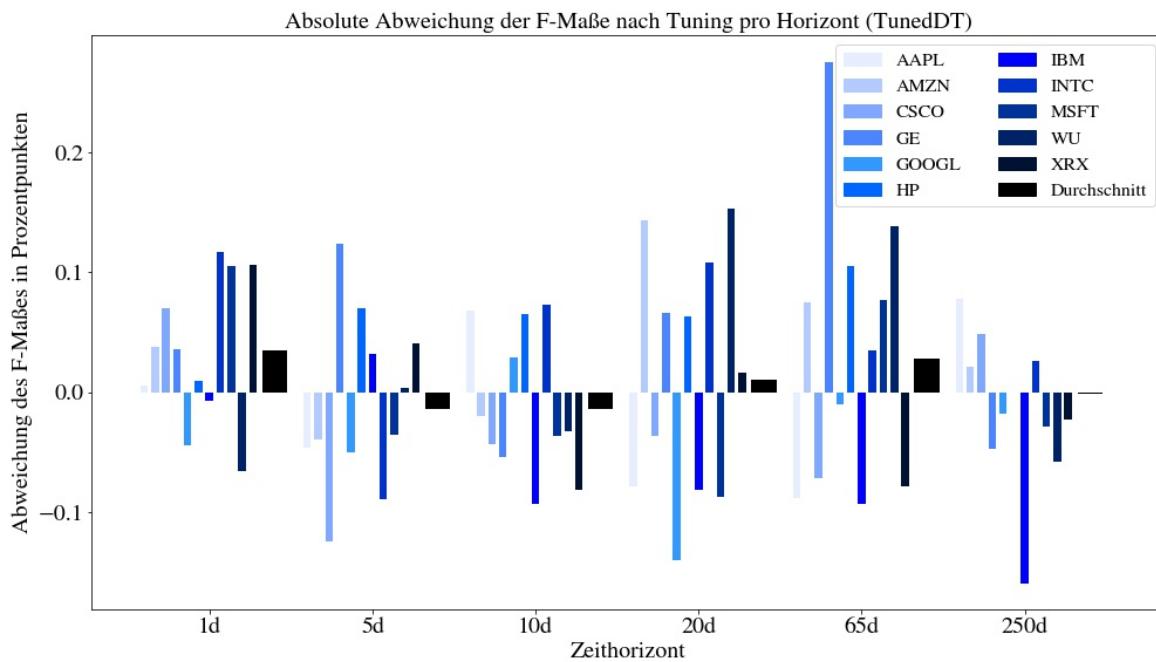


Abbildung 4.12 Einfluss von 10TSCV Hyperparameter-Tuning auf Decision Trees

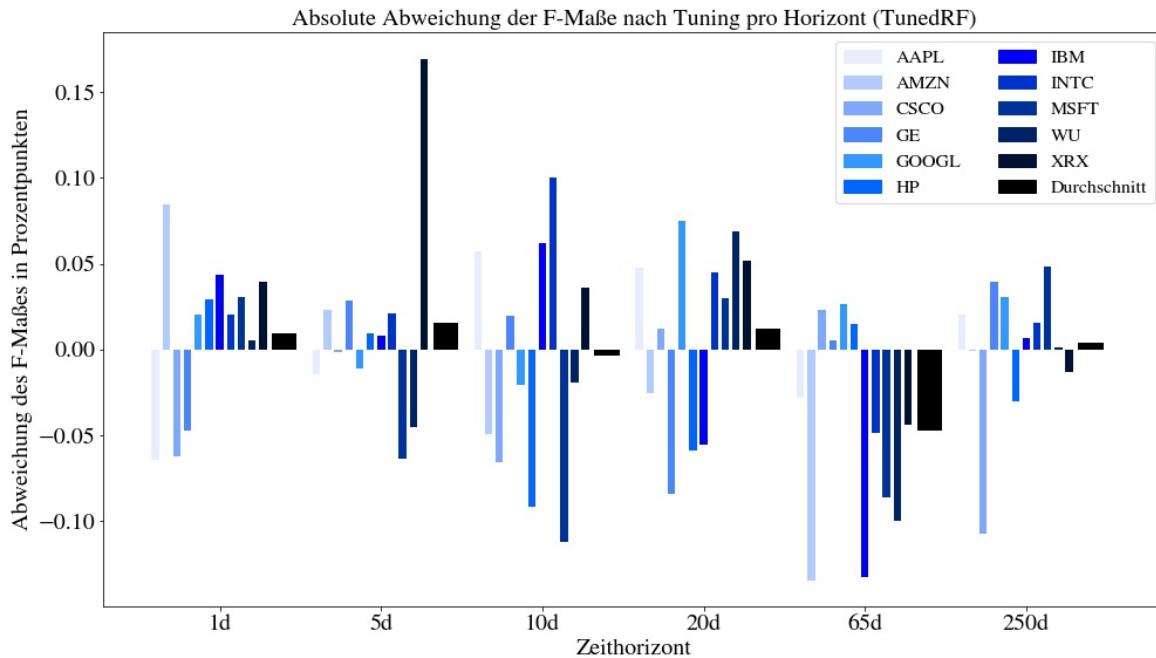


Abbildung 4.13 Einfluss von 10TSCV Hyperparameter-Tuning auf Random Forests

Der Verlauf der F-Maß-Veränderungen unterscheidet sich zwischen den beiden Modelle. Während das F-Maß des Decision Trees auf dem 5d-Horizont durchschnittlich abnimmt (-0,014) und auf dem 65d-Horizont (+0,028) zunimmt, trifft das Gegenteil auf den Random Forest zu (+0,015 und -0,047), wie Abbildung 4.13 zeigt. Es lässt sich kein deutlicher Trend

erkennen, wie das Hyperparameter-Tuning die Klassifikatoren beeinflusst.

Da das Tuning zu variierenden Ergebnissen führt, ist das Overfitting-Problem in diesem Anwendungsfall vergleichsweise weniger präsent, als in Klassifikationen, die durch Tuning nahezu ausschließlich genauer werden. Wie die Statistik über alle Aktiensets und Horizonte zeigt, führt eine Einschränkung der Baumstruktur durch Tuning in ungefähr der Hälfte der Fälle zu einer positiven F-Maß-Veränderung. Auf den insgesamt 66 Kombinationen (11 Aktiensets, 6 Horizonte), auf denen Tuning analysiert wurde, weist der Decision Tree 33 mal (50% der Kombinationen) eine F-Maß-Zunahme und 33 mal (50%) eine Abnahme auf. Der Random Forest liegt 37 mal (56%) höher, die anderen 29 mal (44%) niedriger. Quantifiziert man die F-Maß-Veränderung durch Tuning über alle 66 Kombinationen, so ergibt sich für den Decision Tree ein leicht positiver Wert (+0,007), für Random Forest hingegen ein leicht negativer (-0,004). Wie auch in den bisherigen Auswertungen weist der Random Forest somit erneut eine höhere Robustheit, und eine geringere Abweichung, auf.

Betrachtet man die einzelnen Aktien, so sind auch dort die Auswirkungen von Tuning willkürlich. Das Tuning führt auf den Klassifikatoren zu inkonsistenten Resultaten. So lassen sich beispielsweise für den Decision Tree auf AAPL drei positive Abweichungen (1d, 10d, 250d) und drei negative Abweichungen (5d, 20d, 65d) beobachten. Die AAPL F-Maße des Random Forests steigen und fallen ebenfalls jeweils drei mal, jedoch teilweise auf anderen Horizonten (10d, 20d, 250d steigend und 1d, 5d, 65d fallend). Weder auf dem kürzesten noch auf dem längsten Zeithorizont zeichnet sich ein Trend des Tuning-Einflusses ab.

Betrachtet man nun deutliche F-Maß-Abweichungen von mehr als 10%, so könnte man vermuten, dass deren spezifische Kombinationen aus Horizont und Aktie für den stark positiven, oder stark negativen, Tuning-Einfluss die Ursache ist. Die Regeln, die die Modelle für das jeweilige Aktienset auf dem jeweiligen Horizont erlernen, könnten sich dann besonders gut, oder besonders schlecht, für Tuning eignen – zum Beispiel, weil sie für Overfitting sehr anfällig, oder sehr wenig anfällig, sind.

Es wäre zu erwarten, dass diese Eignung für den Decision Tree und für den Random Forest tendenziell gleich ist. Denn der Random Forest setzt sich aus mehreren Decision Trees zusammen. Sollte also der Decision Tree in Abbildung 4.12 besonders stark oder schwach von Tuning profitieren, so müsste dies auch für den Random Forest in Abbildung 4.13 zutreffen. Es ist zu beachten, dass es Unterschiede im Tuning zwischen den beiden Modellen gibt. Aufgrund begrenzter Rechenressourcen sind die zu vergleichenden Wertekombinationen für den Decision Tree vielfältiger als jene für den Random Forest. Dadurch ist der Random Forest beim Tuning benachteiligt.

Man beachte nun zum Beispiel die deutlichen Abweichungen auf dem Decision Tree: AMZN auf dem 20d-Horizont (+0,143), GE auf dem 65d-Horizont (+0,275) und IBM auf dem 250d-Horizont (-0,159). Für den Random Forest hingegen lauten die entsprechenden Werte: -0,025 (AMZN 20d), +0,005 (GE 65d) und +0,007 (IBM 250d). Das Tuning hat auf die Klassifikatoren unterschiedliche Auswirkungen. Umgekehrt sind einige Ausreißer seitens des Random Forests zum Beispiel XRX auf dem 5d-Horizont (+0,169), MSFT auf dem 10d-Horizont (-0,112) und AMZN auf dem 65d-Horizont (-0,134). Wieder lassen sich diese starken Abweichungen im anderen Klassifikator nicht wiederfinden.

Die Vermutung lässt sich durch die Ergebnisse also nicht bestätigen. Das Tuning führt bei beiden Modellen zu Ausreißern, die im jeweils anderen Modell nicht wiederzufinden sind. Diese deutlichen F-Maß-Abweichungen durch Hyperparameter-Tuning sind demnach mit

den Eigenschaften der jeweiligen Aktien-Horizont-Kombinationen nicht erklärbar. Stattdessen ist der Zufall als Ursache zu nennen. Da der Random Forest seine Basisklassifikatoren auf zufällig ausgewählten Features trainiert, variieren seine Ergebnisse von Natur aus. Ebenfalls erfolgt im Rahmen der Randomized Grid Search die Auswahl der Wertekombinationen der Hyperparameter, die verglichen und optimiert werden, zufällig. Das führt wiederum zu erhöhter Varianz in den Tuning-Auswirkungen.

Neben der bisher behandelten 10TSCV-Methode sind auch die 5TSCV- und die 3TSCV-Methode zur Anwendung gekommen, die die Time-Series Cross Validation auf fünf bzw. drei Teilstichproben durchführen. Weiterhin sind die F-Maß-Veränderungen durch die Kreuzvalidierung mit 5 Stichproben (5fCV) und 3 Stichproben (3fCV) im Anhang angefügt. Sämtliche zusätzliche Diagramme sind im Anhang A.9 zu finden. Auch diese weiteren Tuning-Variationen unterstützen die getroffenen Aussagen.

Als abschließendes Fazit bezüglich Hyperparameter-Tuning lässt sich festhalten, dass dessen Einfluss auf die Genauigkeit der Klassifikatoren stark variiert und sich nicht verallgemeinern lässt. Einerseits verlaufen die F-Maß-Abweichungen bei steigendem Horizont für jeden Klassifikator willkürlich. Andererseits unterscheiden sich diese Verläufe auch für die verschiedenen Klassifikatoren. Zuletzt lässt sich auch, wenn man die F-Maß-Veränderungen auf Aktien-Ebene vergleicht, kein eindeutiges Ergebnis durch Hyperparameter-Tuning feststellen. Ein Investor, der Anlageentscheidungen auf Basis der vorgestellten Klassifikatoren trifft, sollte sich der dargestellten Risiken bewusst sein, und je nach Risikobereitschaft auf Hyperparameter-Tuning verzichten, um die Varianz der F-Maße zu verringern.

4.5 Interpretation der Ergebnisse aus Anwendersicht

Ein Investor, der mit den vier komplexeren Modellen – DT, RF, TunedDT und TunedRF – die Entscheidung trifft, welche Aktien er kauft und welche nicht, und dafür die langfristige zukünftige Richtung des Aktienkurses betrachtet, liegt durchschnittlich bei mehr als 8 von 10 Entscheidungen richtig. Für den 250d-Horizont liegen die F-Maße in Tabelle 4.1 bei über 80%. Investiert er denselben Betrag in jede der 10 Entscheidungen, so kann er Gewinne erzielen. Es ist zu beachten, dass zwar die Richtung der Kursänderung in 8 von 10 Fällen korrekt bestimmt wird, jedoch die absolute oder prozentuale Abweichung, Δ_{Kurs} , unbekannt bleibt. Es besteht die Gefahr, dass die negativen Δ_{Kurs} -Werte der zwei fehlklassifizierten Aktienbewegungen größer sind als jene der korrekt-klassifizierten. Ist das der Fall, würde der Investor – trotz korrekter Vorhersage von 8 von 10 Aktientrends – Verluste erzielen. Wann dieser Fall eintritt und wann nicht, und welche Faktoren für den Erfolg des Investors maßgeblich sind, wird nachfolgend erläutert.

Entscheidet sich der Investor aber dafür, nur in steigende Aktien anzulegen und die fallenden zu vermeiden, dann wird er – unter der Annahme, dass die Aktienmärkte langfristig steigen und positive Δ_{Kurs} -Werte im Durchschnitt höher ausfallen als negative – bei hinreichend hoher Anzahl von Investitionen einen Gewinn erzielen. In der Realität muss diese Annahme aber nicht zutreffen. Ob positive oder negative Δ_{Kurs} -Werte durchschnittlich höher sind, hängt von der Entwicklung des Gesamtmarktes sowie von der Anzahl steigender und fallender Aktien ab. Wenn der Gesamtmarkt konstant bleibt, also ein Index über alle

Aktien weder zu- noch abnimmt, und die Anzahl der steigenden und fallenden Aktien gleich ist, dann würde man im Durchschnitt erwarten, dass der Δ_{Kurs} -Wert in beide Richtungen, positiv oder negativ, gleich ist. Dann erwirtschaftet der Investor mit den komplexen 250d-Klassifikatoren und 8 von 10 richtigen positiven Trendprognosen einen Gewinn, sofern er in eine hinreichend hohe Anzahl an Aktien investiert. Es muss also das große Gesetz der Zahlen greifen, wonach mit steigender Anzahl an Beobachtungen einer Zufallsvariable deren Durchschnitt sich an den Erwartungswert annähert.

Anders verhält es sich in den Extremfällen. Steigen 100% der Aktien, so ist der durchschnittliche positive Δ_{Kurs} -Wert größer als Null, der durchschnittliche negative Δ_{Kurs} -Wert ist dann gleich Null. Dann erzielt der betrachtete Anleger, der nur auf steigenden Kurse setzt, einen Gewinn. Im umgekehrten Fall, wenn 100% der Aktien fallen, so ist der durchschnittliche positive Δ_{Kurs} -Wert gleich Null, der durchschnittliche negative Δ_{Kurs} -Wert ist kleiner Null. Dann erleidet eben dieser Anleger einen Verlust. In der Realität ist davon auszugehen, dass es global gesehen in jedem beliebigen 250-Tage-Horizont sowohl steigende als auch fallende Aktien gibt. Betrachtet man dann den Fall, dass 99% der Aktien steigen, so müssten die fallenden 1% der Aktien deutlich stärker nachgeben, um den Markt-Index konstant zu halten. Also wäre der erwartete positive Δ_{Kurs} -Wert der steigenden Aktien deutlich kleiner als der negative Δ_{Kurs} -Wert der fallenden Aktien. In diesem Fall drohen die 2 Fehlklassifizierungen pro 10 Aktien die Gewinne der 8 korrekten Klassifizierungen auszugleichen, sodass der Investor weder einen Gewinn noch einen Verlust erfährt, oder überzukompensieren, sodass er Verluste erleidet. Im anderen Fall steigen 1% der Aktien sehr stark an, die restlichen 99% fallen leicht. Dann würde die das Gegenteil des genannten Prozederes eintreten, was zum Vorteil des Investors ist.

Es liegt nahe, dass die historische Verteilung von steigenden und fallenden Aktien in den letzten Jahrzehnten zwischen den genannten Extrempunkten lag, und sich diese fortlaufend ändert. Ab einem zu bestimmenden Schwellwert s_{Gewinn} erwartet man dann, dass die Strategie, die 250d-Klassifikatoren anzuwenden und nur in positive Kursänderungen zu investieren, zu Gewinnen führt. Je kleiner der Anteil der steigenden Aktien ist, desto höher ist bei konstantem Gesamtmarkt deren Δ_{Kurs} -Wert im Vergleich zu jenem der fallenden – was zum Vorteil des Investors ist – und vice versa. Wenn nun der Gesamtmarkt nicht konstant ist sondern sich auf- bzw. abwärts bewegt, so führt dies zu einer negativen bzw. positiven Verschiebung des Schwellwerts s_{Gewinn} . Untersucht man die Verteilung von steigenden und fallenden Aktientrends und das gesamte Marktwachstum genauer und prognostiziert diese, zum Beispiel wiederum mit Klassifikatoren, dann lassen sich basierend auf den Erkenntnissen fortgeschrittenere Anlagestrategien formulieren. In Abhängigkeit von dem resultierenden s_{Gewinn} und den erwarteten Marktumständen lässt sich dann beispielsweise bestimmen, ob die Anwendung des 250d-Klassifikators ökonomisch sinnvoll ist oder nicht. Es ist weiterhin zu beachten, dass der Investor nur dann einen beträchtlichen Gewinn erzielen kann, wenn die Anzahl der Aktien, die als positiv klassifiziert und gekauft werden, groß genug ist. Falls der Markt-Index stark steigt und nur sehr wenige Aktien dieses Wachstum begründen, also stark positiv wachsen, und der Klassifikator die Treffergenauigkeit von über 8 von 10 einhält, droht eine Verknappung an steigenden Aktien. Im Extremfall findet der Investor keine oder nur wenige Aktien, die als steigend klassifiziert werden. Dann kann der Anleger das Investitionsrisiko nicht streuen und ist von der Kurssentwicklung weniger Aktien abhängig, ausgleichende Kursschwankungen durch das Gesetz der großen Zahlen bleiben aus. Auch die benötigte Laufzeit zur Klassifikation der Aktien würde dementsprechend verlängert, was möglicherweise zu entgangenen Gewinnen, sicher aber zu erhöhten Ressourcenkosten

führt. Der Investor muss global handeln, damit das Angebot an Aktien möglichst groß ist und seine Strategie funktioniert, und er muss dabei die Effizienz der Klassifizierung sicherstellen. Alternativ kann die Strategie umgedreht werden, sodass der Investor nur auf fallende Aktienkurse setzt. Das bietet sich zum Beispiel in Krisenzeiten an, in denen der Großteil der Aktien und der Markt-Index fallen. Je nach Marktsituation eignet sich eine der beiden Strategien besser, was weiteres Optimierungspotenzial bietet.

5 Schluss

5.1 Zusammenfassung und Ausblick

Diese Arbeit hat die Anwendung von Decision Tree und Random Forest Klassifikatoren in der Finanzdomäne untersucht. Am Beispiel der Aktientrendklassifikation wurden neue Erkenntnisse über das Verhalten, die Stärken und die Schwächen dieser Machine Learning Modelle gewonnen. Vier Baum-basierte Klassifikatoren – Decision Tree und Random Forest, jeweils mit und ohne Hyperparameter-Tuning – wurden dazu mit einem simplen Dummy Klassifikator verglichen. Die Güte der Klassifikation wurde mit dem F-Maß bestimmt, das sich aus dem Time Series Cross-Validation-Verfahren ergibt.

Die Ergebnisse zeigen, dass sowohl die Baum-basierten Klassifikatoren als auch der Dummy auf längeren Zeithorizonten signifikant höhere F-Maße erzielen, als auf kürzeren. Auf kurzfristigen Horizonten liegen die Trefferquoten bei ca. 50%, was der Wahrscheinlichkeit bei zufälligem Raten entspricht. Der Dummy Klassifikator weist hier die höchsten Werte auf. Je länger jedoch der Horizont, desto besser sind die Ergebnisse der Decision Tree und Random Forest Klassifikatoren, sowohl absolut als auch im Vergleich zum Dummy. Für den längsten Horizont, der die Aktienkursbewegung über die nächsten 250 Handelstage erfasst, erzielen die Baum-basierten Klassifikatoren F-Maße von ca. 85%. Damit übertreffen sie den Dummy (74%) signifikant. Daraus folgt, dass Decision Tree und Random Forest Klassifikatoren für den 250d-Horizont sinnvoll zur Aktientrendklassifikation, und allgemeiner in der Finanzdomäne, einsetzbar sind. Auf die kürzeren Horizonte trifft diese Aussage nicht zu. Die Ursache dafür liegt im Random Walk, der vor allem auf kurzfristigen Horizonten stattfindet: Unvorhersehbare Vorkommnisse bestimmen den Aktienkurs. Je länger der prognostizierte Zeitraum, desto geringer ist der Einfluss des Random Walks. Somit werden die Regeln, die die Baum-basierten Klassifikatoren im Training erlernen, bei längerem Zeithorizont zuverlässiger und erreichen höhere F-Maße.

Anschließend hat die Arbeit entdeckt, dass die F-Maße zwischen den einzelnen Aktien erheblich voneinander abweichen. Eine Gruppierung dieser Aktien, basierend auf den Trends ihrer Kurse, ermöglichte eine differenzierte Betrachtung. Auf Aktien, die im Beobachtungszeitraum einen erkennbaren Aufwärts- oder Abwärtstrend verfolgen, zeigen sich bei steigendem Horizont kontinuierlich zunehmende F-Maße, bis diese auf dem 250d-Horizont Maxima erreichen. Aktien mit einem alternierenden Kurs, hingegen, sind am schwierigsten zu klassifizieren: Auf ihnen stagnieren die F-Maße der Klassifikatoren ab dem 65d-Horizont. Diese Resultate pro Gruppe wurden mit einer graphischen Analyse des Kursverlaufes erklärt. Stellt man sich den Klassifikationshorizont als horizontale Linie vor, so führt eine Verlängerung dieses Horizonts bzw. dieser horizontalen Linie zwar bei aufwärts- oder abwärts-verlaufenden Aktien zu eindeutigeren Ergebnissen, nicht jedoch bei alternierenden Aktien, die wellenförmig verlaufen. Es ist kritisch zu beachten, dass diese Ergebnisse auf 11 ausgewählten Aktien aus dem Technologiesektor basieren. In Zukunft sollten weitere Aktien, auch aus anderen Sektoren, auf dieselben Sachverhalte hin untersucht werden. Insbesondere für die Gruppenanalyse sollten weitere Aktien mit den entsprechenden Trends ergänzt

werden, um die Aussagekraft zu steigern.

Eine grundlegende Schwäche des F-Maßes, als Gütemaß für die Klassifikation, hat diese Arbeit am Beispiel des XRX-Datensets identifiziert und analysiert. Da die Berechnung des F-Maßes ausschließlich auf der positiven Klasse basiert, kann dieses den Anwender unter Umständen in die Irre führen. Möchte der Anleger nur auf steigende Aktien setzen, so ist das F-Maß akzeptabel. Will er den Erfolg aber auf allen Klassen messen, weil er zum Beispiel auch auf fallende Kurse setzen möchte, so sollte er zu anderen Gütemaßen greifen. In Realität sollte die Wahl des Gütemaßes deshalb stets von der Handelsstrategie des Anwenders abhängen. Gleiches gilt für den Recall-Precision Tradeoff. Für jeden Investor muss dessen individuelle Risikobereitschaft bekannt sein, damit der jeweils optimale Klassifikator bestimmt werden kann.

Weiterhin hat sich ergeben, dass die Klassen bei steigendem Horizont tendenziell einseitiger verteilt sind. Findet man auf dem 1d-Horizont häufig eine ausgewogene 50%-50%-Verteilung zwischen der positiven und der negativen Klasse vor, so schwindet dieses Gleichgewicht zunehmend auf höheren Horizonten. Diese Erkenntnis ermöglichte Schlüsse auf die Lernfähigkeiten der verschiedenen Klassifikatoren. Vergleiche zwischen F-Maßen bei zunehmender Einseitigkeit der Klassen und F-Maßen bei gleichbleibender Klassenverteilung ergaben eine höhere Anpassungsfähigkeit der Decision Tree und Random Forest Klassifikatoren. Im Gegensatz zum Dummy Klassifikator sind diese nämlich dazu in der Lage, selbst bei – über mehrere Horizonte – gleichbleibender Klassen-Verteilung höhere F-Maße zu erzielen. Für den Dummy lässt sich dies nicht beobachten. Er erreicht nur dann höhere F-Maße, wenn auch die Klassenverteilung einseitiger wird. Das zeugt von einer niedrigen Lernfähigkeit. Dank ihrer höheren Anpassungsfähigkeit sind also die Baum-basierten Klassifikatoren zur Aktienklassifikation besser geeignet, als der Dummy. Anknüpfende Forschung sollte Möglichkeiten zur Wiederherstellung des Klassengleichgewichts, wie dem Over- und Undersampling, erproben. Damit kann ein Klassifikator auf beiden Klassen, die zuvor über- oder unter-repräsentiert waren, ausreichend trainiert werden und möglicherweise höhere F-Maße erreichen.

Als Ursache für die höhere Flexibilität von Decision Tree und Random Forest wurde deren Gewichtung der Features, in Abhängigkeit vom zu klassifizierenden Zeithorizont, ausgemacht. Mittels gefärbten Punktewolken wurde graphisch und numerisch gezeigt, dass diese Modelle auf längeren Horizonten langfristige Features signifikant höher gewichten, als kurzfristige Features. Der Dummy hingegen nimmt keine solche Gewichtung der Features vor, sondern fällt seine Entscheidung alleine auf Grundlage der Klassenverteilung.

Der Random Forest hat sich mehrmals als robuster als der Decision Tree herausgestellt. Manipulationen der (Hyper-)Parameter, der Trainingsdaten oder anderer Umweltfaktoren, führen im Random Forest stets zu gedämpfteren Abweichungen, als im Decision Tree. Untersuchungen der Feature-Gewichtungen, sowie der Einflüsse von Features Extraction und Tuning, unterstützen diese Aussage. Die höhere Robustheit des Random Forests wurde auf dessen Eigenschaft zurückgeführt, dass er mehrere Decision Trees kombiniert. Zum Beispiel gewichtet der Random Forest zwar langfristige Features für einen langen Horizont tendenziell höher. Aber dieser Zusammenhang ist deutlich schwächer ausgeprägt, als beim Decision Tree. Hinzu kommt, dass der Random Forest ab dem 10d-Horizont das "ohlc_avg"-Feature – das keine langfristige Kursentwicklung, sondern nur den jeweiligen Tag abbildet – von allen Features am stärksten gewichtet. Dabei kam die Vermutung auf, dass dieses Feature eine

besonders hohe Aussagekraft besitzt, weil es als Durchschnitt von vier anderen Features berechnet wurde und diese somit zusammenfasst. Solche zusammenfassenden Features waren für die Horizonte von 1d bis 250d nicht vorhanden. Deshalb wurde vorgeschlagen, in Zukunft für sämtliche Horizonte kombinierende, und möglicherweise aussagekräftigere, Features zu produzieren. Zu deren Berechnung sollten, neben dem Durchschnitt, auch andere Kombinationsmethoden ausprobiert werden. Zum Beispiel eignet sich zur Kombination des gleitenden Durchschnitts und der Volatilität die Multiplikation besser, da somit die Volatilität, trotz ihrer absolut kleineren Größe, stärker miteinfließt, als es im Durchschnitt der Fall ist. Falls solche Kombinationsfeatures, ähnlich wie das kurzfristige "ohlc_avg"-Feature, in den längerfristigen Horizonten entsprechend hoch in den Modellen gewichtet werden, so können diese neuen Features die Genauigkeiten der Klassifikatoren steigern.

Im Gegensatz zur referenzierten Literatur, hat diese Arbeit den Einfluss von Feature Extraction explizit untersucht. Dazu wurden die F-Maße, die unter Anwendung von Features Extraction entstanden sind, jenen F-Maßen gegenübergestellt, welche ohne Feature Extraction zustande kamen. Aus den ursprünglich sechs enthaltenen Features pro Datenset wurden durch Berechnungen weitere 25 Features generiert. Die zusätzlichen technischen Features, darunter der gleitende Durchschnitt und die Volatilität der Tagesschlusspreise, wurden für jeden Zeithorizont separat berechnet.

Feature Extraction führt im Durchschnitt auf allen Baum-basierten Klassifikatoren zu höheren F-Maßen. Der Dummy ist davon, wie erwartet, nicht betroffen, da er seine Entscheidungen nur auf die Klassenverteilungen stützt, ohne Beachtung der Features. Eine Differenzierung der Ergebnisse nach Horizonten lässt erkennen, dass Feature Extraction auf den kurz- und langfristigen Horizonten für die Klassifikatoren den größten Nutzen hat. Auf dem 250d-Horizont profitieren die Modelle am stärksten. Dabei erzielt der Decision Tree deutlich höhere F-Maß-Steigerungen, als der Random Forest. Das liegt daran, dass er die Feature-Gewichtung stärker am Zeithorizont ausrichtet und somit auch von den zusätzlichen technischen Features mehr profitiert, als der robustere Random Forest. Die langfristigen Features helfen den Klassifikatoren langfristige Trends zu erkennen und somit höhere F-Maße zu erreichen. Auf mittelfristigen Horizonten hingegen führt Feature Extraction teilweise zu niedrigeren F-Maßen. Diese Beobachtung ist konsistent mit den Ergebnissen aus der Feature-Gewichtung: Während für die kurz- und langfristigen Features klare Trends bei steigenden Zeithorizont zu erkennen sind, lassen die mittelfristigen Features keine klaren Muster erkennen. So haben selbst auf den mittelfristigen Zeithorizonten die langfristigen Features klar dominiert. Dadurch hat Feature Extraction mittelfristig einen schwächeren, teilweise negativen, Einfluss auf die Klassifikatoren, als auf kurz- oder langfristigen.

Die Arbeit hat Hyperparameter-Tuning, im Anwendungsfall der Aktientrendklassifikation, als nur bedingt hilfreich beurteilt. Unterschiedliche Verfahren, sowohl Time Series Cross-Validation als auch Kreuzvalidierungen mit verschiedenen Anzahlen an Epochen, haben gleichermaßen zu dem Ergebnis geführt, dass Tuning einen unvorhersehbaren Einfluss auf den Erfolg von Decision Tree und Random Forest Klassifikatoren hat. In 50% der untersuchten Fälle ist nach Tuning eine positive F-Maß-Änderung zu beobachten, in den anderen 50% eine negative. Ein konservativer Anwender sollte deshalb auf Hyperparameter-Tuning verzichten, um die Varianz der F-Maße niedriger zu halten. Die durchschnittliche F-Maß-Veränderung, über alle betrachteten Fälle hinweg, demonstriert erneut die Robustheit des Random Forests: Seine F-Maße sinken um 0,3%, während die des Decision Trees um 0,7% steigen. Hierbei ist zu beachten, dass das Random Forest-Tuning, aufgrund begrenzter

Rechenressourcen, weniger Werte zur Optimierung des F-Maßes verglichen hat, und somit gegenüber dem Decision Tree benachteiligt ist.

Zuletzt hat die Interpretation der Ergebnisse aus Anwendersicht gezeigt, dass ein Investor, der die vorgestellten Klassifikatoren auf dem 250d-Horizont anwendet, zwar in mehr als acht von zehn Fällen den Aktientrend korrekt prognostiziert. Aber ohne die Ausmaße der zukünftigen Aufwärts- und Abwärtsbewegungen zu kennen, sind Gewinne nur unter bestimmten Voraussetzungen möglich. Anknüpfende Forschung könnte weitere Klassifikatoren in die Anlageentscheidung einfließen lassen, um beispielsweise die Entwicklung des Gesamtmarktes abzuschätzen. Mit einer solchen Ergänzung würden makroökonomische Entwicklungen berücksichtigt, die – zusätzlich zum historischen Aktienkurs – ein weiterer wichtiger Faktor für den Aktientrend sind. Denn fällt der Gesamtmarkt stark, so könnte der Investor zum Beispiel – sofern der Klassifikator über Soft Voting entscheidet – nur auf die sichersten 10% der positiven Instanzen setzen, und vice versa.

Die Ergebnisse befürworten die langfristige Aktientrendklassifikation mittels Decision Trees und Random Forests als vielversprechendes Forschungsgebiet. Nicht nur wurden F-Maße von über 85% erreicht. Gleichzeitig wurden zentrale Eigenschaften dieser Modelle auf mehreren Dimensionen – darunter der Zeithorizont und das Datenset – kritisch untersucht. Insbesondere die Erforschung der Fähigkeit von Baum-basierten Klassifikatoren, Features abhängig vom zu klassifizierenden Zeithorizont zu gewichten, stellt einen wichtigen Fortschritt dar. Am Maßstab eines Dummy Klassifikator wurde die Lernfähigkeit dieser Modelle festgemacht. Unterschiede zwischen dem Decision Tree und dem Random Forest wurden aufgezeigt. Basierend auf den gewonnenen Erkenntnissen folgten begründete Vorschläge, wie zukünftige Forschung die Aktientrendklassifikation weiter verbessern kann.

A Anhang

A.1 Aktienverläufe absolut

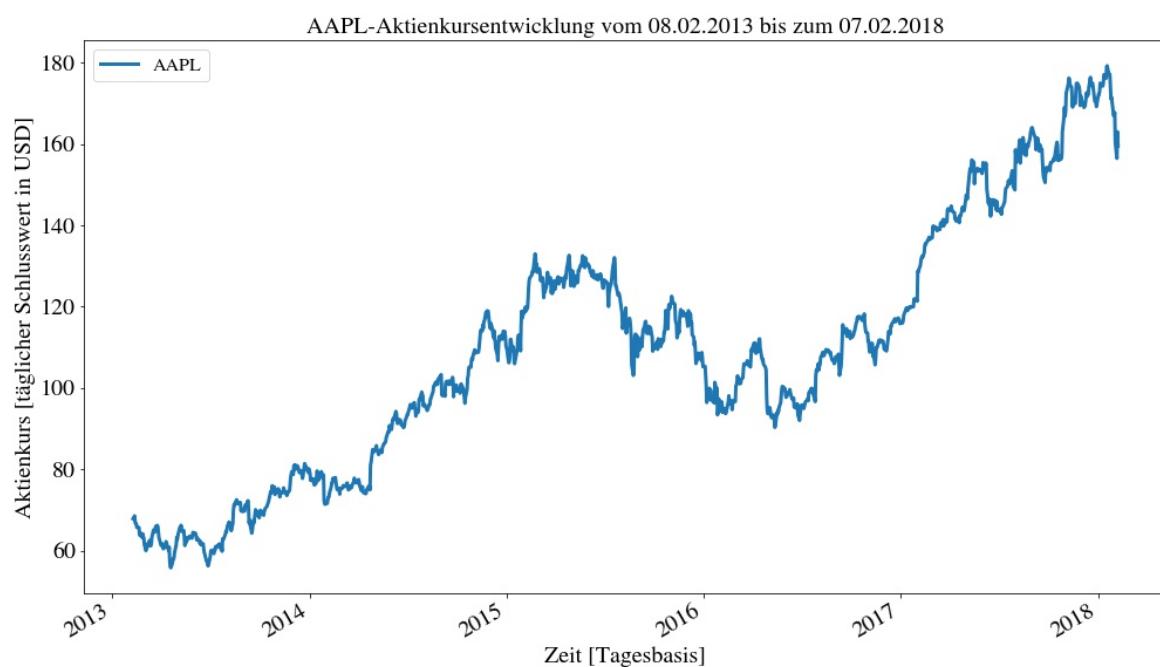


Abbildung A.1 AAPL-Aktienkurs zwischen 2013 und 2018

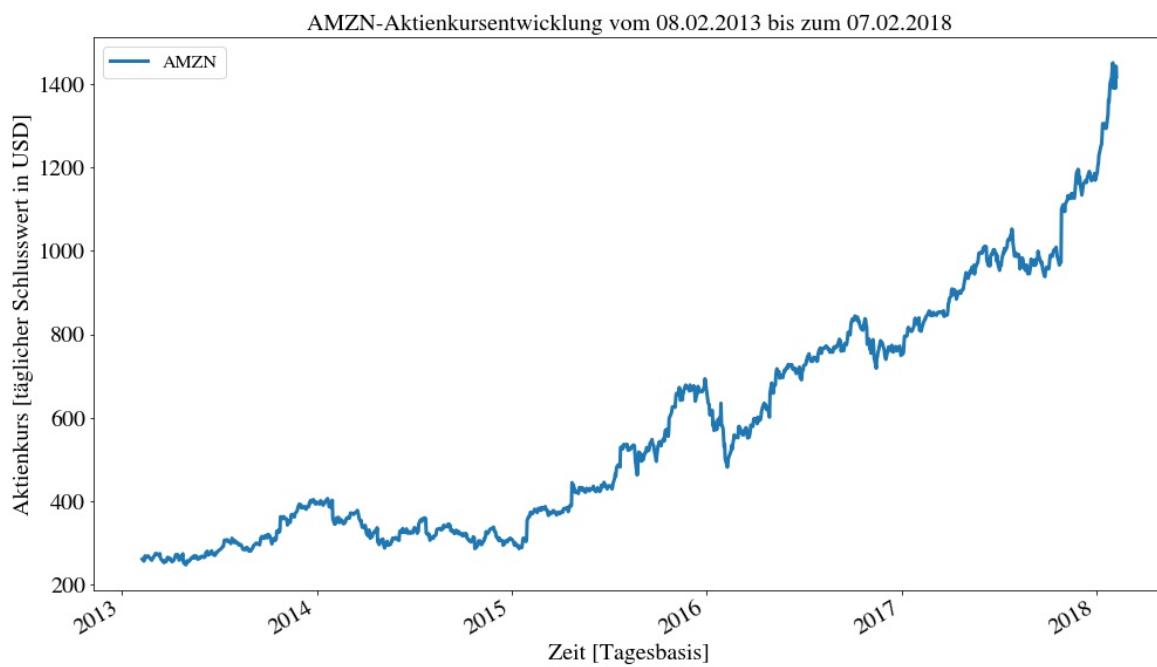


Abbildung A.2 AMZN-Aktienkurs zwischen 2013 und 2018



Abbildung A.3 CSCO-Aktienkurs zwischen 2013 und 2018

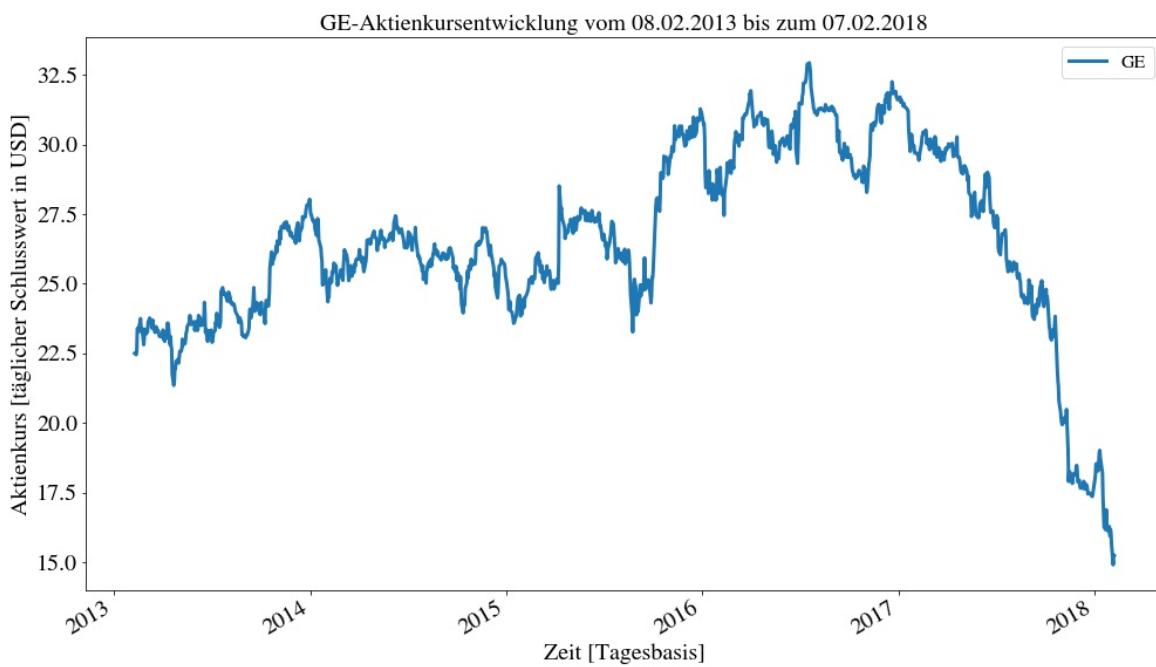


Abbildung A.4 GE-Aktienkurs zwischen 2013 und 2018

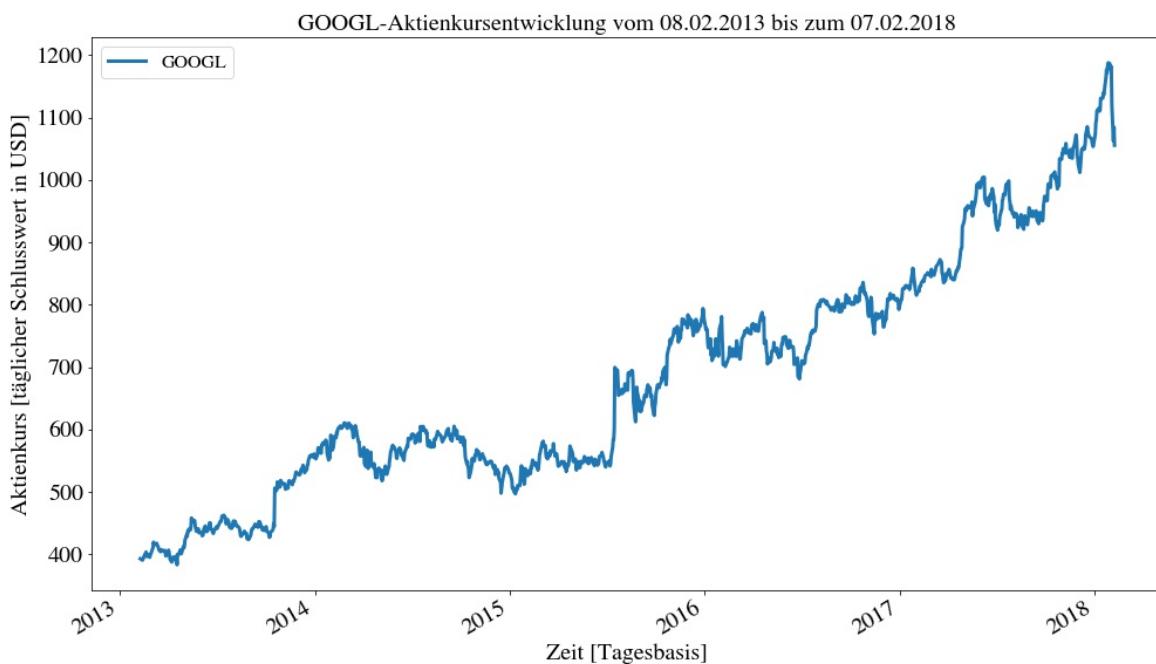


Abbildung A.5 GOOGL-Aktienkurs zwischen 2013 und 2018

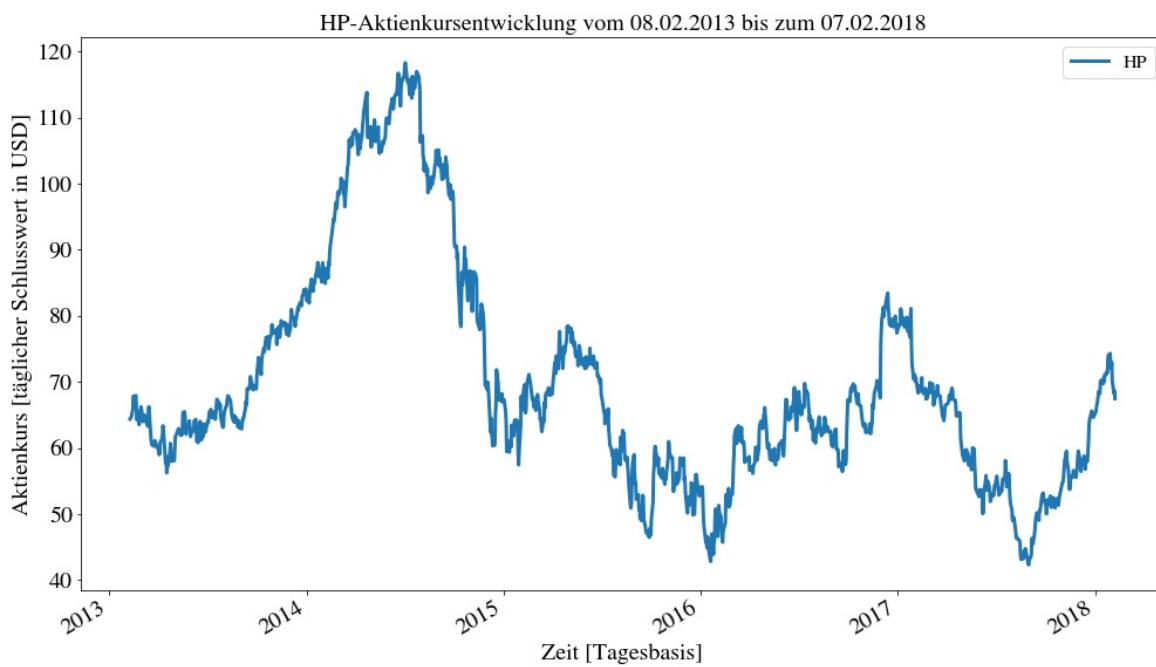


Abbildung A.6 HP-Aktienkurs zwischen 2013 und 2018



Abbildung A.7 IBM-Aktienkurs zwischen 2013 und 2018

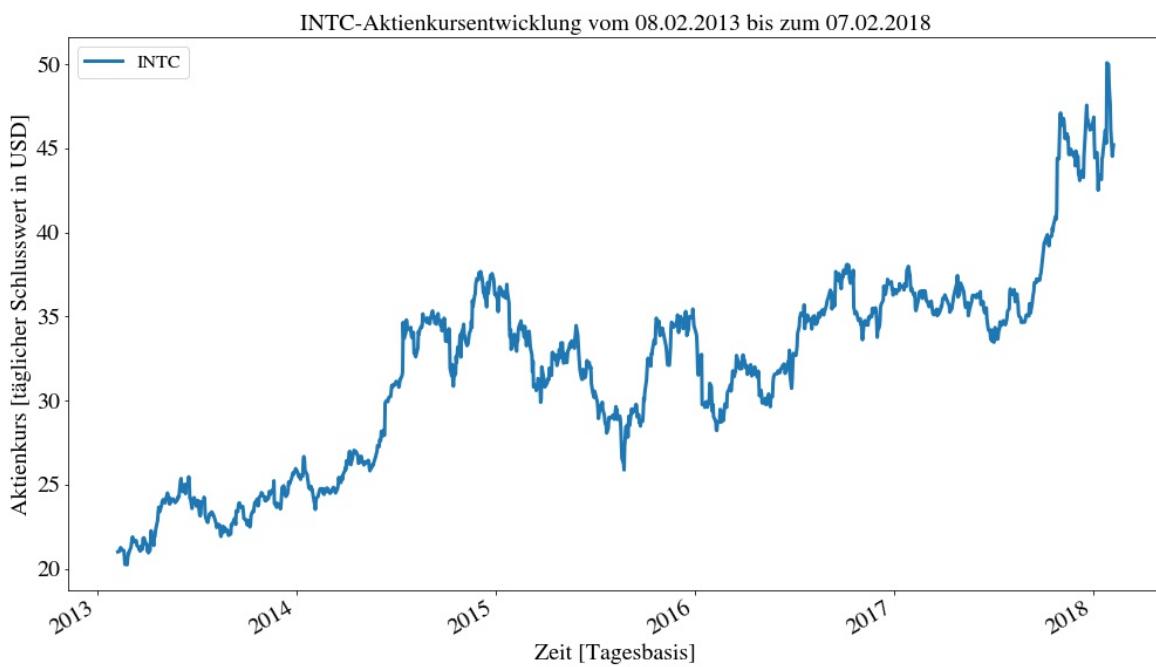


Abbildung A.8 INTC-Aktienkurs zwischen 2013 und 2018



Abbildung A.9 MSFT-Aktienkurs zwischen 2013 und 2018

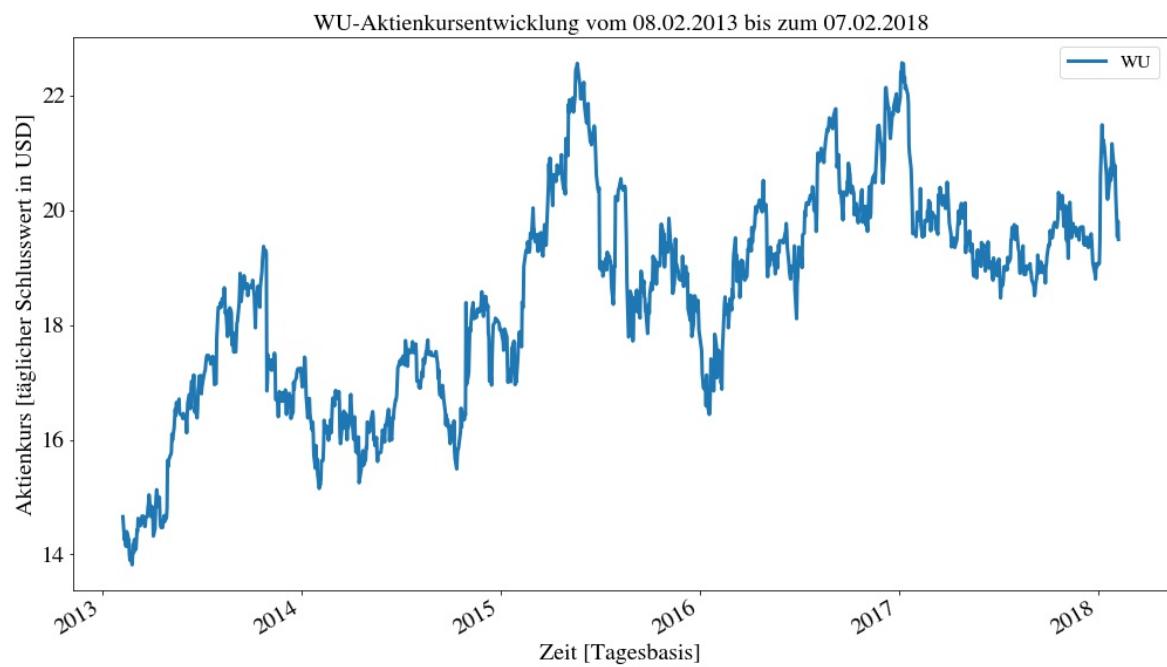


Abbildung A.10 WU-Aktienkurs zwischen 2013 und 2018

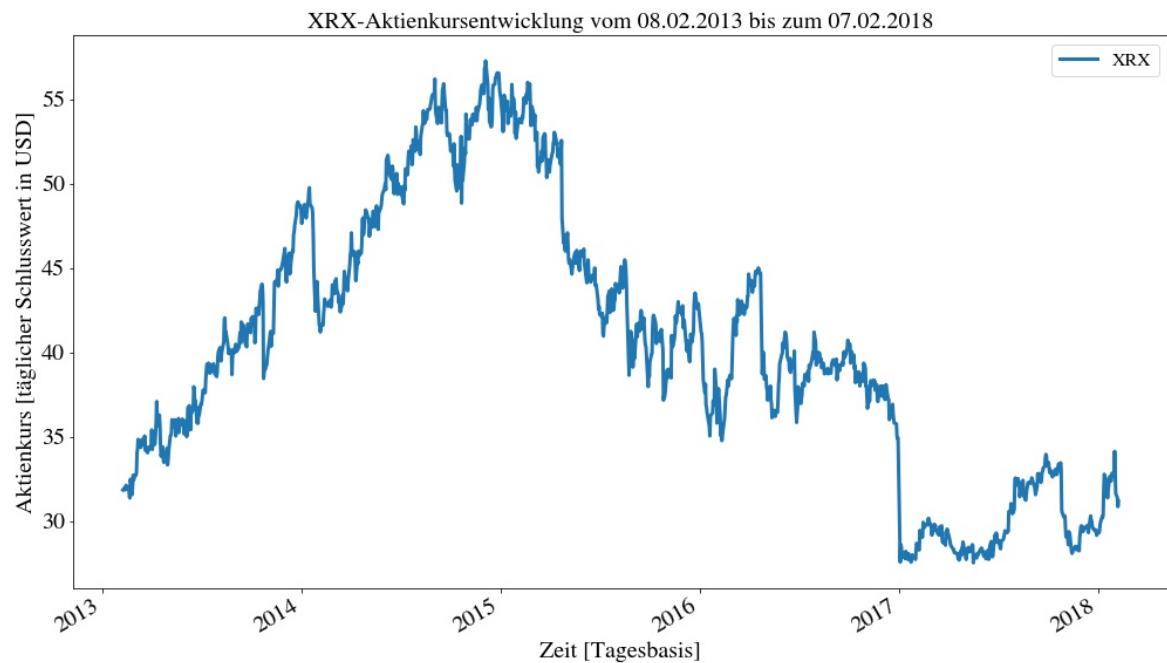


Abbildung A.11 XRX-Aktienkurs zwischen 2013 und 2018

A.2 Auflistung der Features

Die folgenden 31 Features stehen den Klassifikatoren pro Instanz, zur Bestimmung der Klasse, zur Verfügung. Die ersten 6 Features stammen direkt aus den jeweiligen Datensets, die restlichen 25 Features werden im Feature Extraction-Schritt generiert.

- | | |
|---------------------|----------------------|
| 1. open | 17. volatility_10d |
| 2. high | 18. ma_10d |
| 3. low | 19. momentum_10d |
| 4. close | 20. return_past_20d |
| 5. volume | 21. volatility_20d |
| 6. month | 22. ma_20d |
| 7. ohlc_avg | 23. momentum_20d |
| 8. return_past_1d | 24. return_past_65d |
| 9. volatility_1d | 25. volatility_65d |
| 10. ma_1d | 26. ma_65d |
| 11. momentum_1d | 27. momentum_65d |
| 12. return_past_5d | 28. return_past_250d |
| 13. volatility_5d | 29. volatility_250d |
| 14. ma_5d | 30. ma_250d |
| 15. momentum_5d | 31. momentum_250d |
| 16. return_past_10d | |

A.3 Klassen-Häufigkeitsverteilungen der Aktien pro Horizont

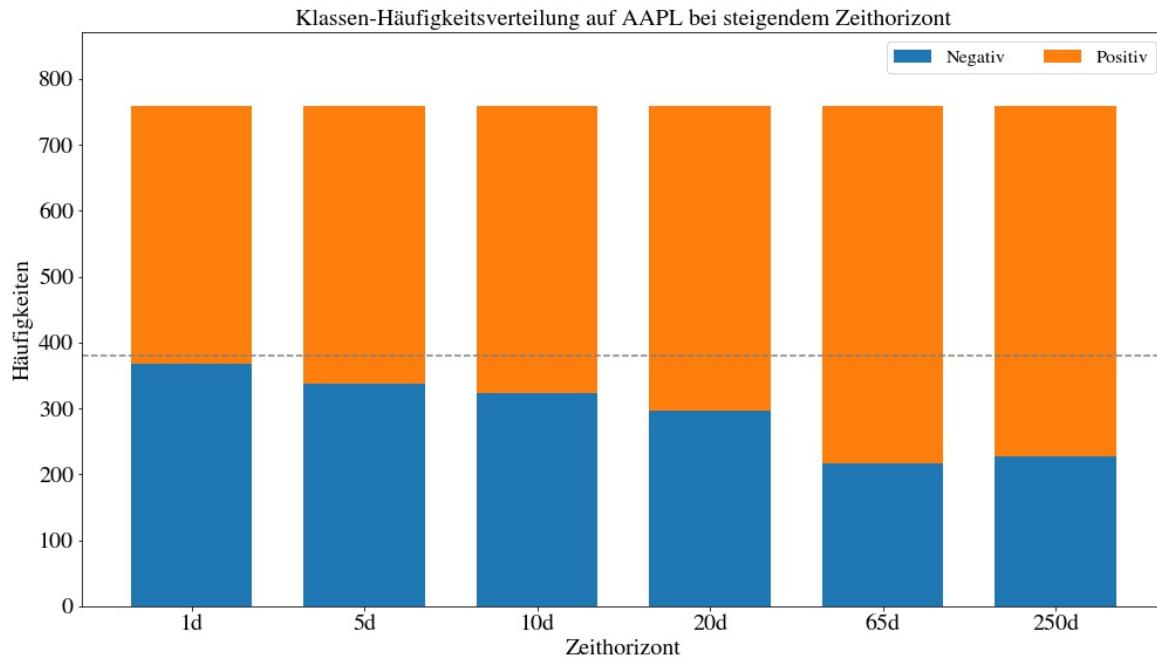


Abbildung A.12 AAPL-Klassenverteilungen pro Horizont

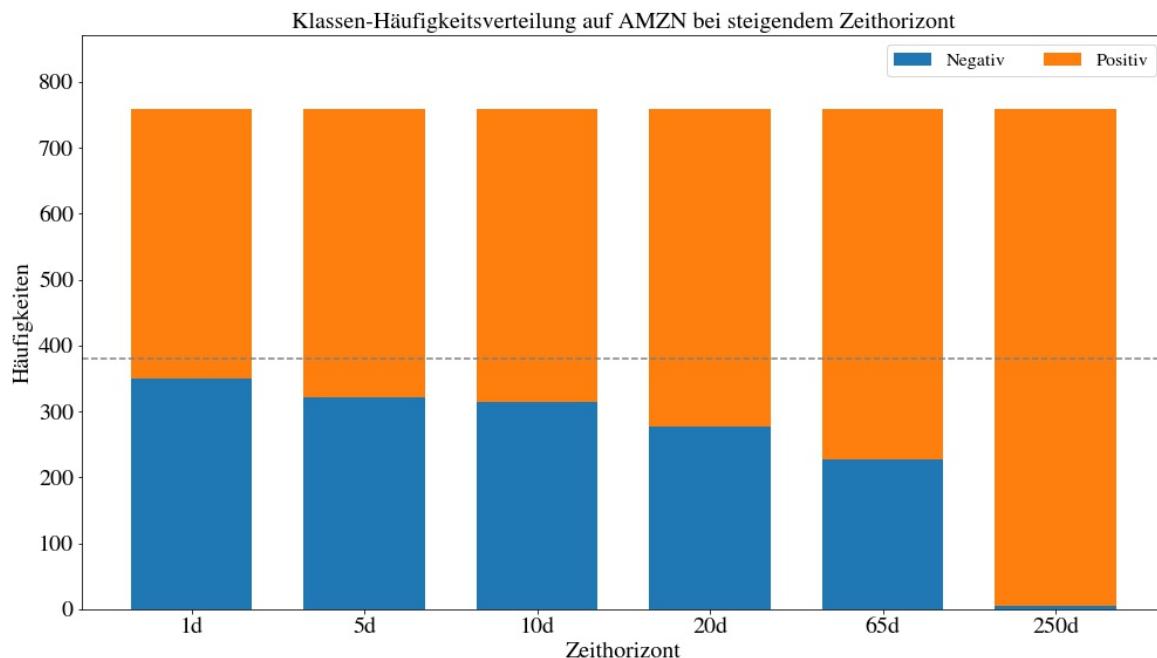


Abbildung A.13 AMZN-Klassenverteilungen pro Horizont

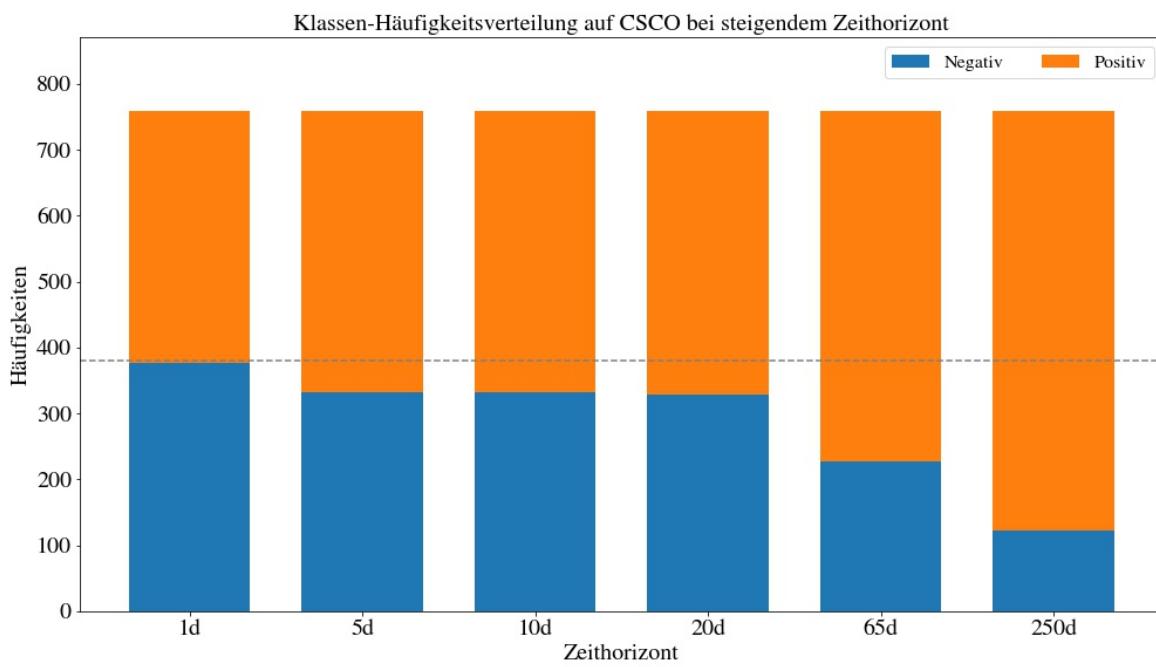


Abbildung A.14 CSCO-Klassenverteilungen pro Horizont

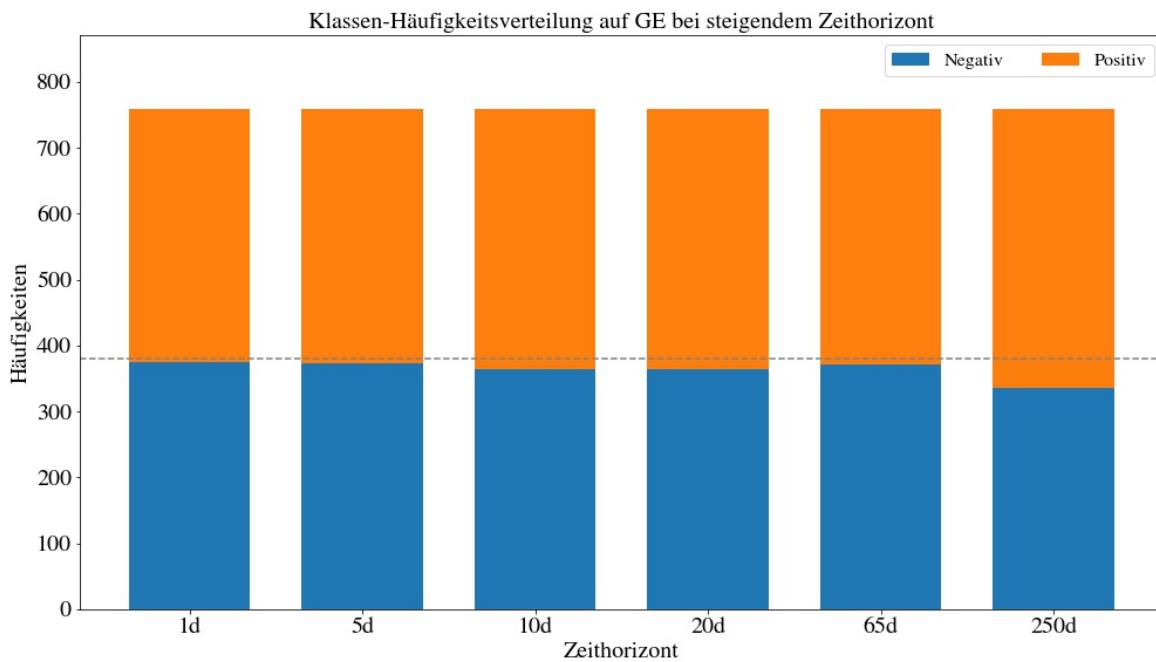


Abbildung A.15 GE-Klassenverteilungen pro Horizont

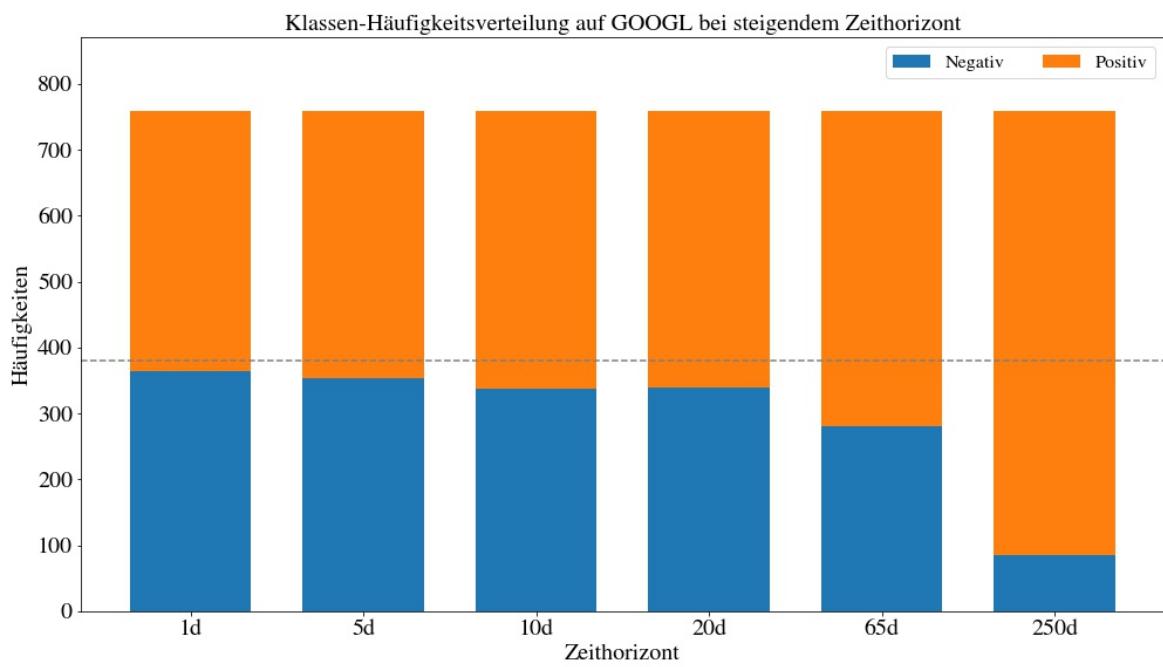


Abbildung A.16 GOOGL-Klassenverteilungen pro Horizont

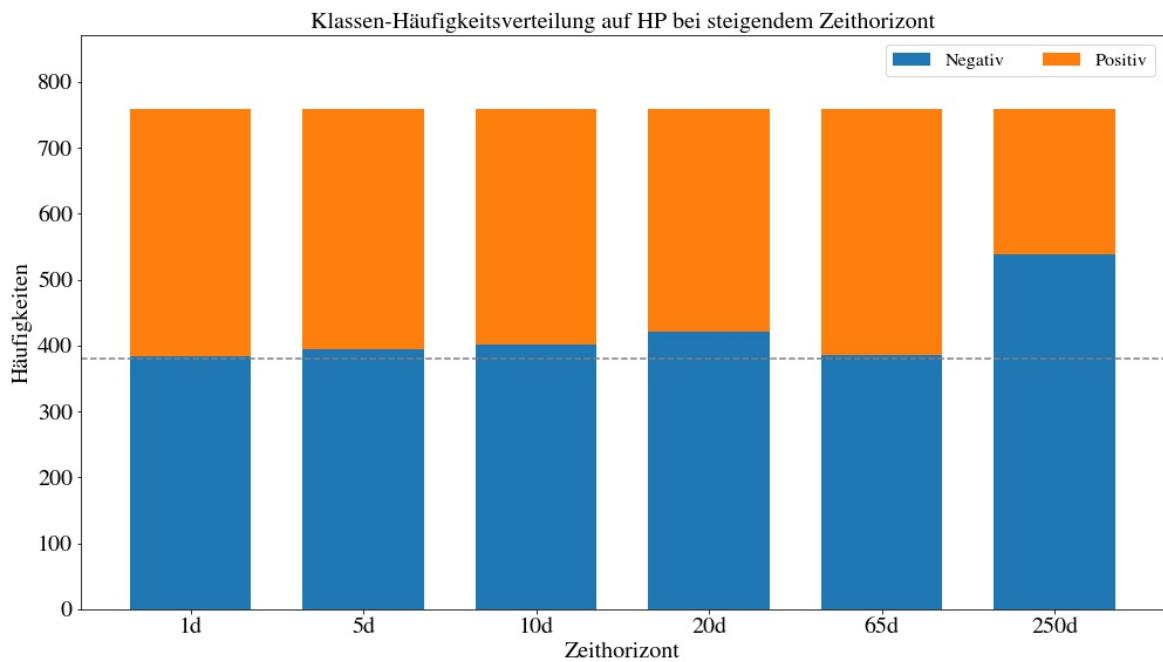


Abbildung A.17 HP-Klassenverteilungen pro Horizont

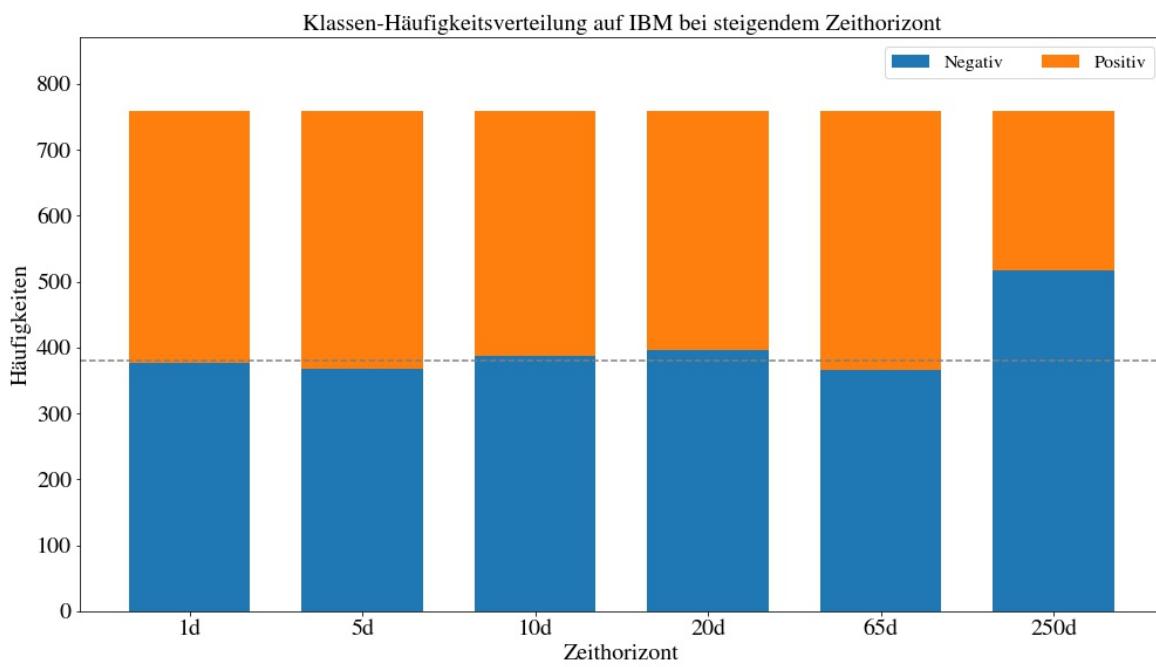


Abbildung A.18 IBM-Klassenverteilungen pro Horizont

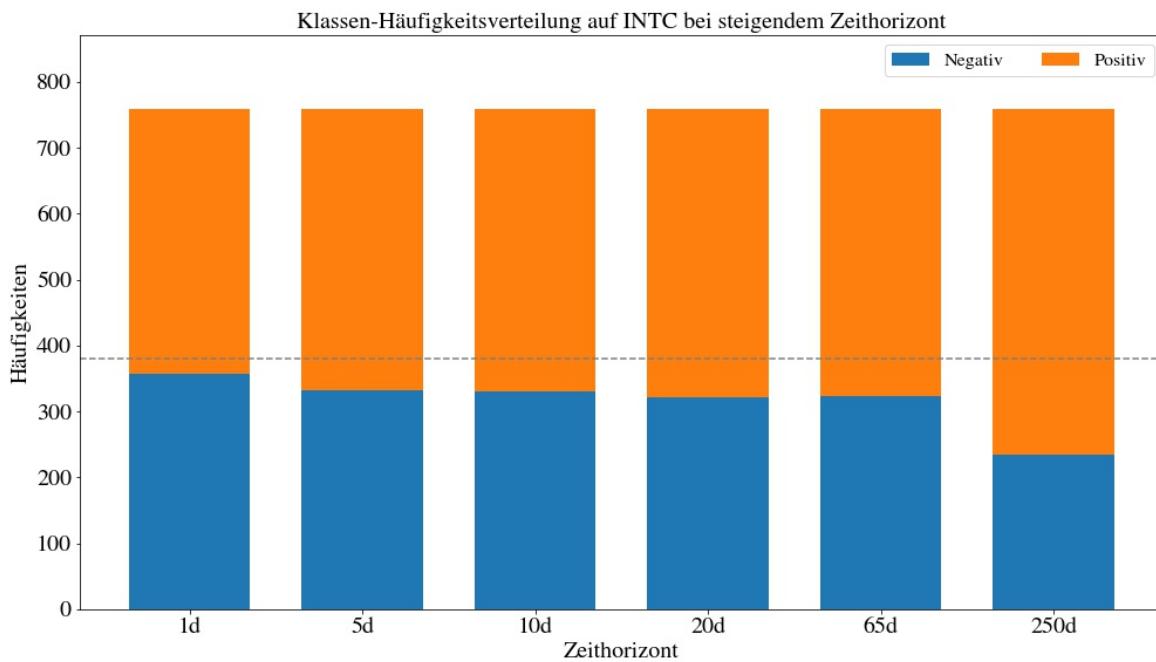


Abbildung A.19 INTC-Klassenverteilungen pro Horizont

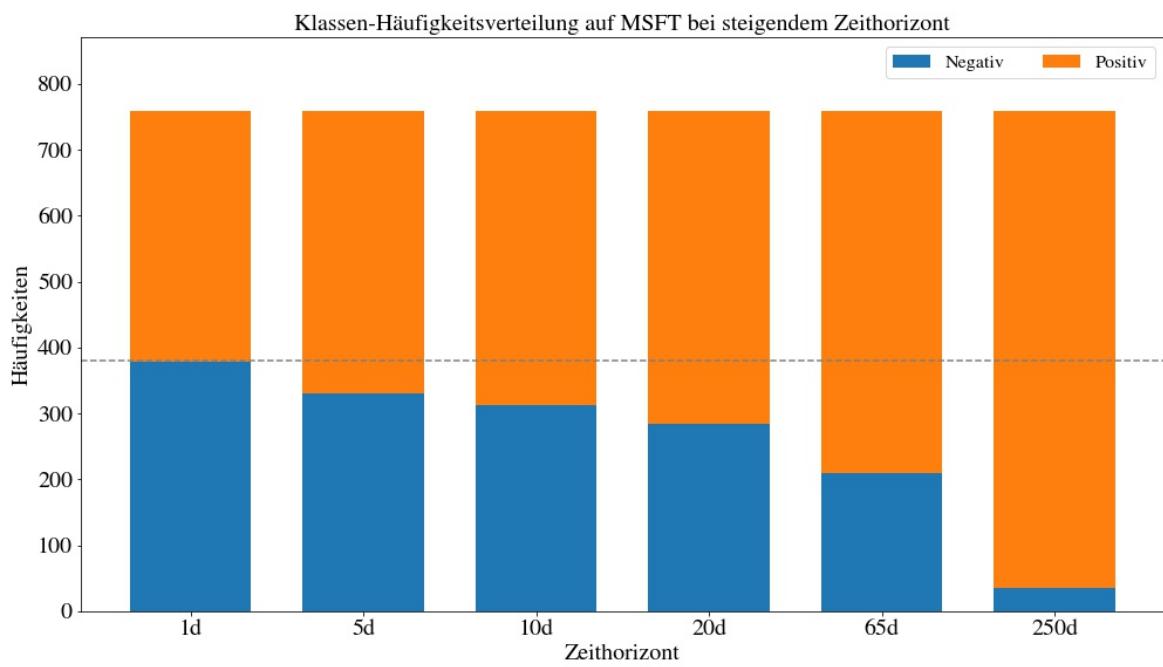


Abbildung A.20 MSFT-Klassenverteilungen pro Horizont

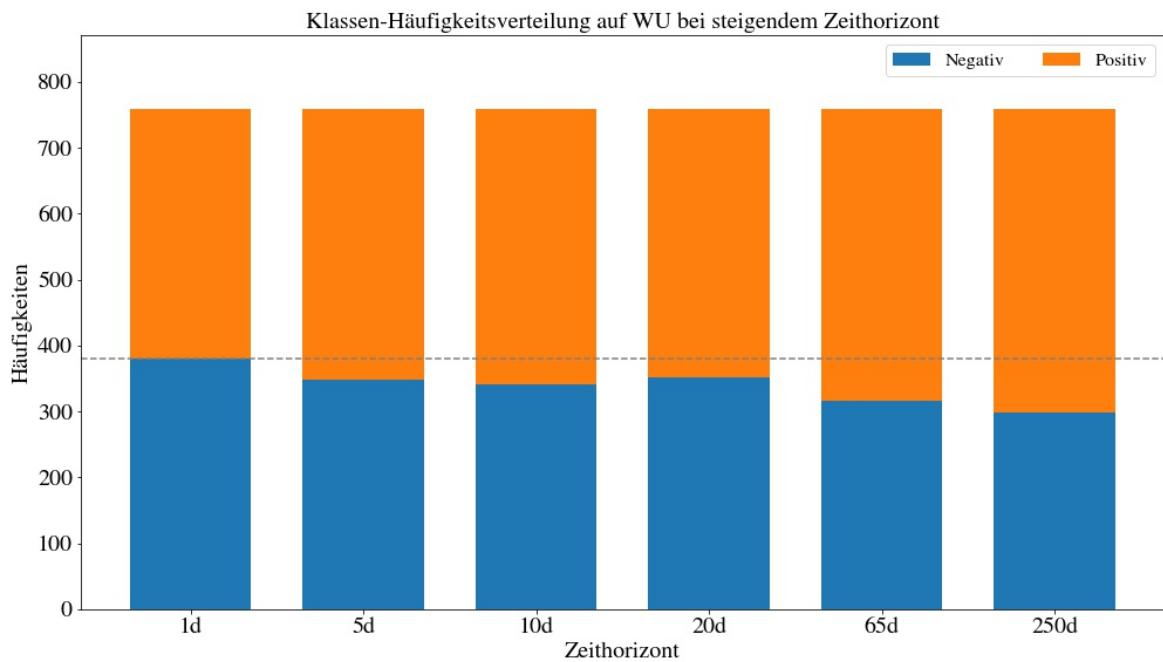


Abbildung A.21 WU-Klassenverteilungen pro Horizont

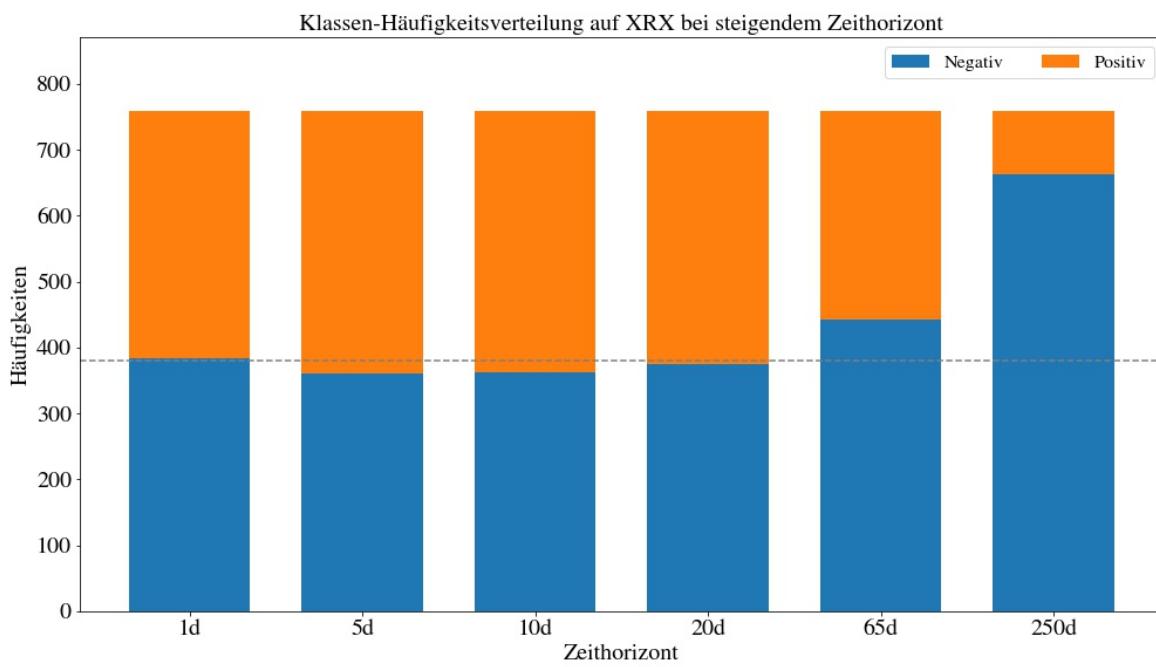


Abbildung A.22 XRX-Klassenverteilungen pro Horizont

A.4 Auswertungen der Klassifikatoren: F-Maße pro Aktie pro Horizont als Diagramme

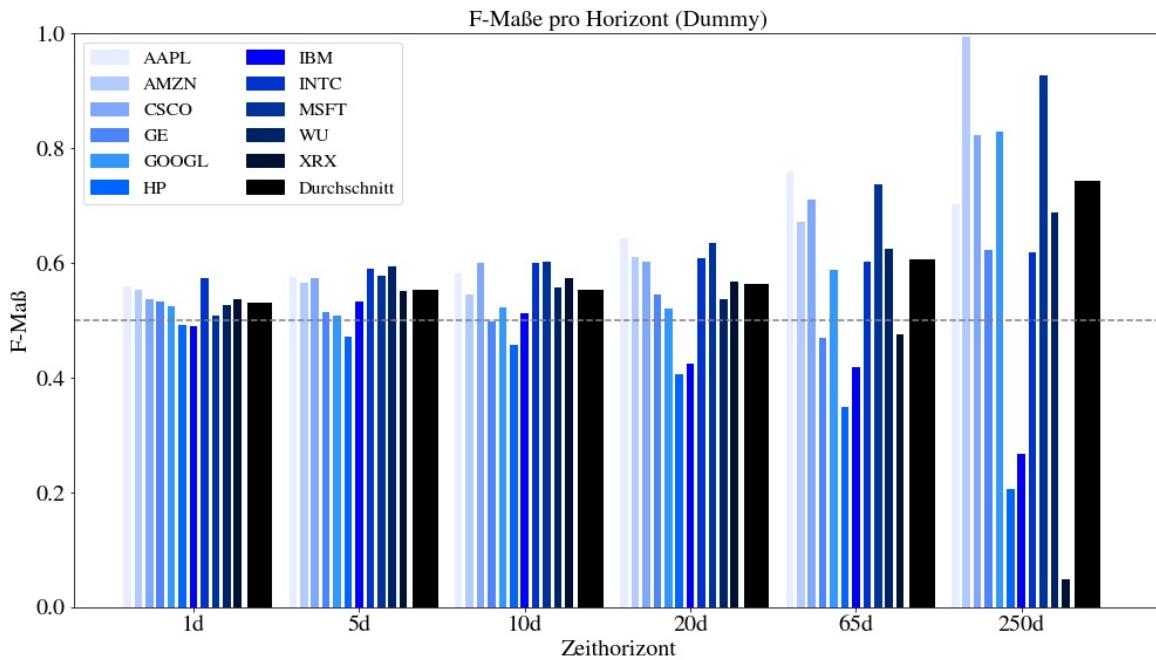


Abbildung A.23 Dumm: F-Maße pro Horizont pro Aktie

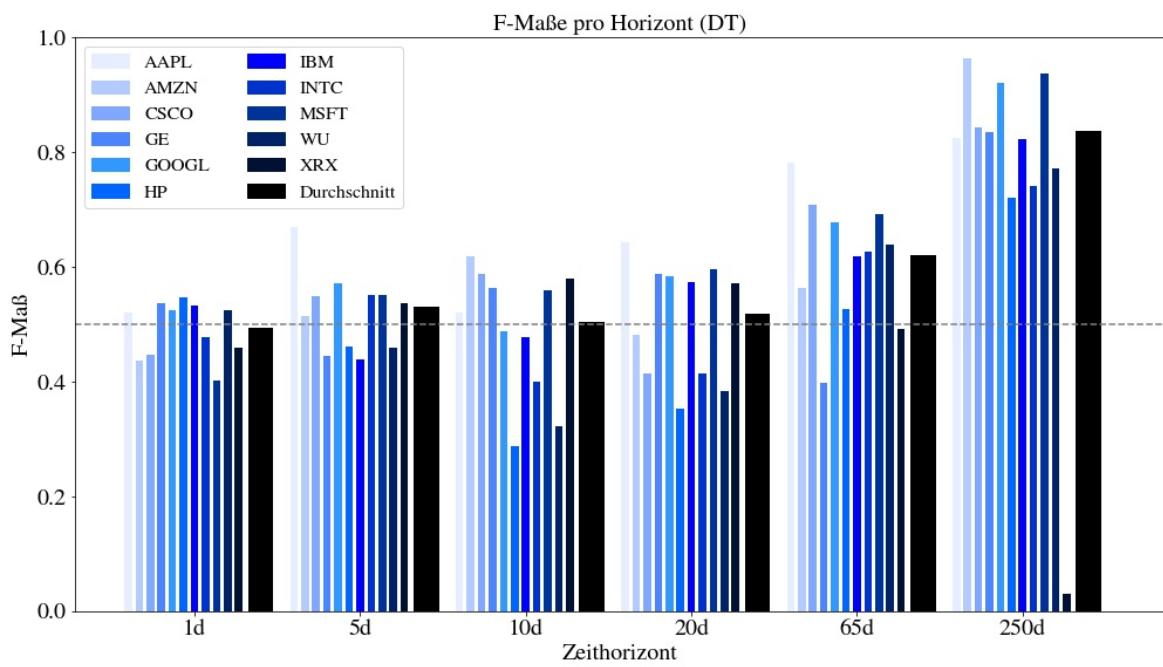


Abbildung A.24 Decision Tree: F-Maße pro Horizont pro Aktie

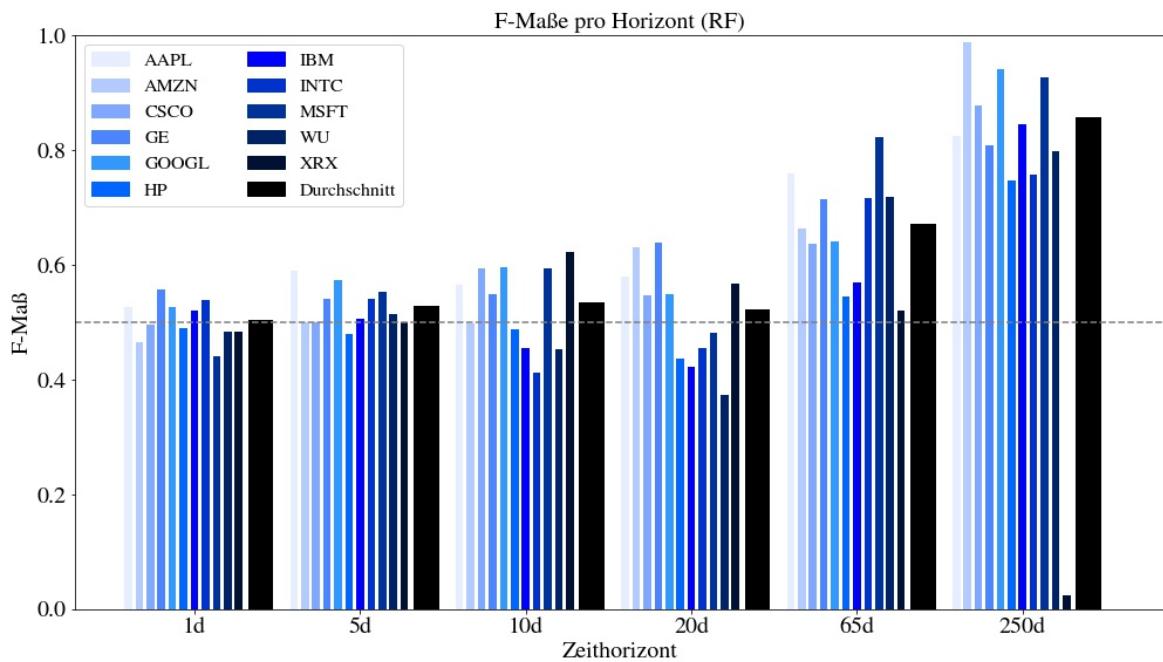


Abbildung A.25 Random Forest: F-Maße pro Horizont pro Aktie

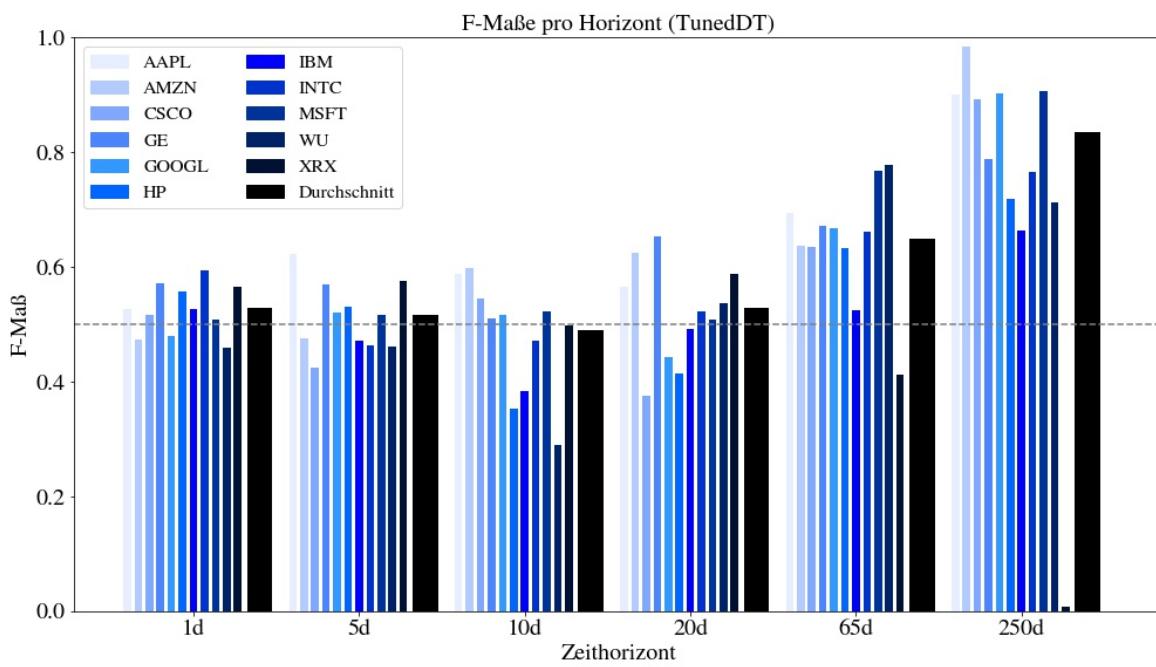


Abbildung A.26 Decision Tree mit Tuning: F-Maße pro Horizont pro Aktie

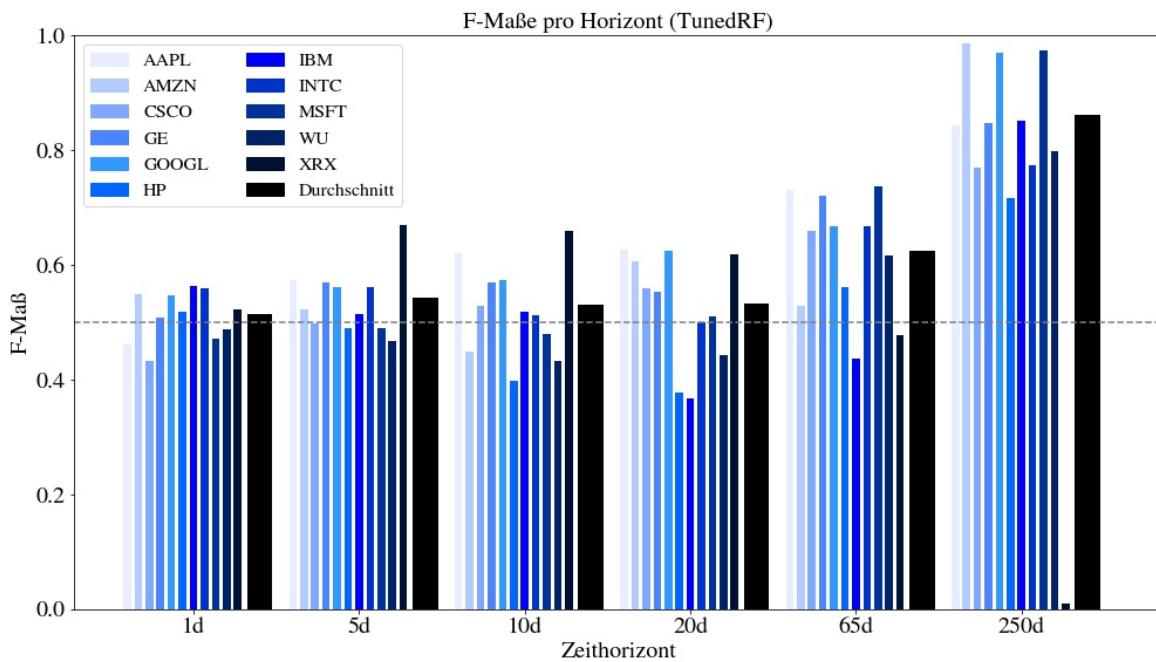


Abbildung A.27 Random Forest mit Tuning: F-Maße pro Horizont pro Aktie

A.5 Auswertungen der Klassifikatoren: Precision, Recall und F-Maß in Tabellen

Tabelle A.1 Konfusionsmetriken im Durchschnitt über alle Datensets

Durchschnitt über alle Datensets							
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,519	0,531	0,538	0,540	0,582	0,743
	DT	0,514	0,553	0,551	0,540	0,632	0,845
	RF	0,515	0,553	0,571	0,542	0,658	0,885
	TunedDT	0,515	0,547	0,556	0,550	0,637	0,836
	TunedRF	0,521	0,539	0,556	0,550	0,618	0,879
Recall	Dummy	0,543	0,576	0,570	0,589	0,633	0,741
	DT	0,476	0,509	0,463	0,497	0,611	0,828
	RF	0,493	0,505	0,501	0,503	0,686	0,832
	TunedDT	0,544	0,488	0,438	0,508	0,662	0,835
	TunedRF	0,507	0,548	0,507	0,519	0,632	0,845
F-Maß	Dummy	0,531	0,552	0,554	0,563	0,607	0,742
	DT	0,495	0,530	0,503	0,518	0,621	0,836
	RF	0,504	0,528	0,534	0,522	0,672	0,858
	TunedDT	0,529	0,516	0,490	0,528	0,649	0,835
	TunedRF	0,514	0,543	0,530	0,534	0,625	0,862

Tabelle A.2 Konfusionsmetriken auf AAPL

AAPL							
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,533	0,536	0,546	0,599	0,686	0,656
	DT	0,515	0,576	0,575	0,602	0,802	0,905
	RF	0,508	0,565	0,578	0,647	0,691	0,867
	TunedDT	0,524	0,571	0,589	0,547	0,685	0,879
	TunedRF	0,529	0,533	0,588	0,622	0,671	0,884
Recall	Dummy	0,588	0,621	0,621	0,695	0,850	0,756
	DT	0,525	0,797	0,474	0,692	0,764	0,756
	RF	0,545	0,616	0,551	0,525	0,843	0,785
	TunedDT	0,528	0,685	0,587	0,584	0,704	0,925
	TunedRF	0,410	0,624	0,659	0,633	0,805	0,807
F-Maß	Dummy	0,559	0,575	0,581	0,643	0,759	0,703
	DT	0,520	0,669	0,520	0,644	0,782	0,824
	RF	0,526	0,589	0,564	0,580	0,759	0,824
	TunedDT	0,526	0,623	0,588	0,565	0,694	0,901
	TunedRF	0,462	0,575	0,622	0,628	0,732	0,844

Tabelle A.3 Konfusionsmetriken auf AMZN

AMZN							
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,533	0,582	0,580	0,686	0,762	1,000
	DT	0,555	0,606	0,603	0,646	0,649	1,000
	RF	0,562	0,599	0,583	0,667	0,706	1,000
	TunedDT	0,571	0,565	0,628	0,657	0,670	1,000
	TunedRF	0,559	0,612	0,528	0,672	0,634	1,000
Recall	Dummy	0,573	0,550	0,513	0,550	0,599	0,987
	DT	0,359	0,448	0,635	0,385	0,496	0,928
	RF	0,397	0,428	0,436	0,599	0,625	0,976
	TunedDT	0,405	0,410	0,572	0,597	0,607	0,969
	TunedRF	0,541	0,455	0,392	0,552	0,454	0,975
F-Maß	Dummy	0,552	0,565	0,545	0,611	0,670	0,993
	DT	0,436	0,515	0,618	0,483	0,562	0,963
	RF	0,465	0,499	0,499	0,631	0,663	0,988
	TunedDT	0,474	0,476	0,599	0,625	0,637	0,984
	TunedRF	0,550	0,522	0,450	0,606	0,529	0,987

Tabelle A.4 Konfusionsmetriken auf CSCO

CSCO							
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,527	0,563	0,571	0,555	0,657	0,812
	DT	0,496	0,599	0,631	0,506	0,786	0,862
	RF	0,483	0,545	0,649	0,615	0,728	0,966
	TunedDT	0,480	0,547	0,613	0,534	0,710	0,886
	TunedRF	0,480	0,553	0,578	0,665	0,726	0,853
Recall	Dummy	0,547	0,586	0,631	0,659	0,774	0,832
	DT	0,406	0,506	0,550	0,349	0,644	0,825
	RF	0,509	0,463	0,547	0,493	0,565	0,805
	TunedDT	0,561	0,347	0,490	0,291	0,576	0,898
	TunedRF	0,395	0,455	0,487	0,483	0,604	0,704
F-Maß	Dummy	0,537	0,574	0,600	0,602	0,710	0,822
	DT	0,447	0,549	0,587	0,413	0,708	0,843
	RF	0,496	0,501	0,594	0,547	0,636	0,878
	TunedDT	0,518	0,425	0,544	0,377	0,636	0,892
	TunedRF	0,433	0,499	0,528	0,560	0,659	0,771

Tabelle A.5 Konfusionsmetriken auf GE

GE							
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,514	0,495	0,471	0,485	0,459	0,557
	DT	0,520	0,524	0,554	0,558	0,412	0,849
	RF	0,526	0,567	0,557	0,627	0,692	0,738
	TunedDT	0,511	0,541	0,544	0,619	0,639	0,721
	TunedRF	0,509	0,576	0,568	0,580	0,702	0,809
Recall	Dummy	0,552	0,536	0,529	0,620	0,482	0,706
	DT	0,555	0,388	0,574	0,620	0,383	0,821
	RF	0,590	0,516	0,543	0,650	0,740	0,894
	TunedDT	0,651	0,601	0,480	0,692	0,710	0,868
	TunedRF	0,509	0,563	0,571	0,530	0,740	0,891
F-Maß	Dummy	0,532	0,515	0,498	0,544	0,470	0,623
	DT	0,537	0,446	0,564	0,587	0,397	0,835
	RF	0,556	0,540	0,550	0,638	0,715	0,808
	TunedDT	0,573	0,569	0,510	0,653	0,672	0,788
	TunedRF	0,509	0,569	0,570	0,554	0,720	0,848

Tabelle A.6 Konfusionsmetriken auf GOOGL

GOOGL							
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,518	0,524	0,571	0,576	0,621	0,975
	DT	0,533	0,553	0,572	0,580	0,689	0,947
	RF	0,545	0,577	0,630	0,628	0,730	0,990
	TunedDT	0,593	0,539	0,601	0,561	0,659	0,935
	TunedRF	0,560	0,587	0,627	0,642	0,721	0,992
Recall	Dummy	0,534	0,495	0,482	0,473	0,559	0,721
	DT	0,514	0,592	0,426	0,586	0,667	0,897
	RF	0,511	0,570	0,564	0,488	0,572	0,897
	TunedDT	0,402	0,505	0,454	0,366	0,676	0,875
	TunedRF	0,536	0,541	0,531	0,609	0,623	0,951
F-Maß	Dummy	0,525	0,509	0,523	0,520	0,588	0,829
	DT	0,523	0,572	0,488	0,583	0,678	0,921
	RF	0,527	0,573	0,595	0,550	0,642	0,941
	TunedDT	0,479	0,522	0,517	0,443	0,667	0,904
	TunedRF	0,548	0,563	0,575	0,625	0,668	0,971

Tabelle A.7 Konfusionsmetriken auf HP

HP						
Horizont		1d	5d	10d	20d	65d
Precision	Dummy	0,469	0,449	0,435	0,375	0,368
	DT	0,497	0,457	0,341	0,329	0,436
	RF	0,476	0,433	0,453	0,368	0,469
	TunedDT	0,474	0,510	0,378	0,388	0,583
	TunedRF	0,477	0,458	0,398	0,324	0,473
Recall	Dummy	0,515	0,498	0,482	0,440	0,331
	DT	0,607	0,467	0,249	0,379	0,666
	RF	0,506	0,539	0,531	0,534	0,653
	TunedDT	0,675	0,555	0,331	0,448	0,691
	TunedRF	0,571	0,527	0,397	0,451	0,688
F-Maß	Dummy	0,491	0,472	0,457	0,405	0,349
	DT	0,547	0,462	0,288	0,352	0,527
	RF	0,490	0,480	0,489	0,436	0,546
	TunedDT	0,557	0,532	0,353	0,415	0,632
	TunedRF	0,520	0,490	0,397	0,377	0,561

Tabelle A.8 Konfusionsmetriken auf IBM

IBM						
Horizont		1d	5d	10d	20d	65d
Precision	Dummy	0,532	0,512	0,548	0,456	0,527
	DT	0,511	0,529	0,477	0,479	0,597
	RF	0,503	0,503	0,451	0,418	0,602
	TunedDT	0,511	0,507	0,435	0,491	0,509
	TunedRF	0,556	0,472	0,493	0,380	0,470
Recall	Dummy	0,452	0,554	0,482	0,398	0,348
	DT	0,557	0,376	0,476	0,714	0,641
	RF	0,539	0,510	0,461	0,429	0,540
	TunedDT	0,542	0,440	0,342	0,494	0,542
	TunedRF	0,571	0,566	0,545	0,357	0,408
F-Maß	Dummy	0,489	0,532	0,513	0,425	0,419
	DT	0,533	0,440	0,476	0,574	0,618
	RF	0,520	0,507	0,456	0,423	0,569
	TunedDT	0,526	0,471	0,383	0,492	0,525
	TunedRF	0,564	0,515	0,518	0,368	0,437

Tabelle A.9 Konfusionsmetriken auf INTC

ITNC		1d	5d	10d	20d	65d	250d
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,544	0,557	0,563	0,554	0,519	0,626
	DT	0,525	0,580	0,528	0,516	0,751	0,819
	RF	0,541	0,622	0,561	0,480	0,683	0,877
	TunedDT	0,531	0,568	0,582	0,539	0,634	0,831
	TunedRF	0,527	0,565	0,597	0,528	0,629	0,863
Recall	Dummy	0,605	0,625	0,645	0,675	0,716	0,612
	DT	0,437	0,526	0,321	0,347	0,538	0,676
	RF	0,535	0,477	0,326	0,434	0,755	0,667
	TunedDT	0,675	0,391	0,397	0,508	0,691	0,712
	TunedRF	0,594	0,557	0,449	0,476	0,713	0,701
F-Maß	Dummy	0,573	0,589	0,601	0,609	0,602	0,619
	DT	0,477	0,552	0,399	0,415	0,627	0,741
	RF	0,538	0,540	0,413	0,456	0,717	0,758
	TunedDT	0,594	0,463	0,472	0,523	0,661	0,767
	TunedRF	0,559	0,561	0,513	0,501	0,668	0,773

Tabelle A.10 Konfusionsmetriken auf MSFT

MSFT		1d	5d	10d	20d	65d	250d
Horizont		1d	5d	10d	20d	65d	250d
Precision	Dummy	0,503	0,541	0,575	0,601	0,691	0,950
	DT	0,474	0,598	0,650	0,672	0,759	0,963
	RF	0,476	0,606	0,729	0,596	0,876	0,950
	TunedDT	0,520	0,578	0,662	0,599	0,855	0,948
	TunedRF	0,494	0,583	0,657	0,620	0,859	0,954
Recall	Dummy	0,515	0,618	0,633	0,671	0,789	0,906
	DT	0,350	0,510	0,490	0,535	0,634	0,911
	RF	0,409	0,508	0,500	0,403	0,776	0,904
	TunedDT	0,497	0,466	0,432	0,441	0,698	0,871
	TunedRF	0,450	0,421	0,379	0,434	0,645	0,997
F-Maß	Dummy	0,509	0,577	0,603	0,634	0,736	0,927
	DT	0,403	0,551	0,559	0,595	0,691	0,936
	RF	0,440	0,553	0,593	0,481	0,823	0,927
	TunedDT	0,508	0,516	0,523	0,508	0,768	0,908
	TunedRF	0,471	0,489	0,481	0,511	0,737	0,975

Tabelle A.11 Konfusionsmetriken auf WU

WU						
Horizont		1d	5d	10d	20d	65d
Precision	Dummy	0,504	0,573	0,544	0,527	0,560
	DT	0,549	0,515	0,500	0,438	0,613
	RF	0,549	0,582	0,576	0,481	0,669
	TunedDT	0,498	0,528	0,503	0,565	0,755
	TunedRF	0,531	0,514	0,586	0,534	0,580
Recall	Dummy	0,551	0,616	0,571	0,546	0,704
	DT	0,503	0,414	0,238	0,341	0,668
	RF	0,432	0,460	0,373	0,306	0,774
	TunedDT	0,426	0,411	0,204	0,511	0,802
	TunedRF	0,452	0,430	0,344	0,379	0,661
F-Maß	Dummy	0,526	0,593	0,557	0,536	0,624
	DT	0,525	0,459	0,323	0,384	0,640
	RF	0,483	0,514	0,453	0,374	0,718
	TunedDT	0,459	0,462	0,290	0,537	0,778
	TunedRF	0,489	0,469	0,433	0,443	0,618

Tabelle A.12 Konfusionsmetriken auf XRX

XRX						
Horizont		1d	5d	10d	20d	65d
Precision	Dummy	0,536	0,495	0,507	0,478	0,378
	DT	0,485	0,523	0,535	0,606	0,435
	RF	0,515	0,524	0,558	0,476	0,410
	TunedDT	0,507	0,559	0,514	0,522	0,323
	TunedRF	0,500	0,525	0,535	0,508	0,373
Recall	Dummy	0,536	0,619	0,661	0,699	0,641
	DT	0,437	0,550	0,632	0,541	0,563
	RF	0,455	0,479	0,705	0,703	0,714
	TunedDT	0,641	0,596	0,482	0,674	0,571
	TunedRF	0,548	0,923	0,860	0,794	0,661
F-Maß	Dummy	0,536	0,550	0,574	0,568	0,476
	DT	0,460	0,536	0,579	0,572	0,491
	RF	0,483	0,500	0,623	0,568	0,521
	TunedDT	0,566	0,577	0,498	0,588	0,413
	TunedRF	0,523	0,669	0,659	0,620	0,477

A.6 F-Maße pro Aktie ohne Feature Extraction

Tabelle A.13 F-Maße pro Aktie ohne Feature Extraction

F-Maße ohne Feature Extracation	Horizont	Dummy	DT	RF	TunedDT	TunedRF
TECH GROUP	1d	0.531	0.489	0.487	0.499	0.472
	5d	0.552	0.522	0.514	0.540	0.545
	10d	0.554	0.505	0.498	0.484	0.513
	20d	0.563	0.547	0.555	0.547	0.554
	65d	0.607	0.632	0.657	0.627	0.630
	250d	0.742	0.779	0.816	0.784	0.832
AAPL	1d	0.559	0.560	0.553	0.619	0.536
	5d	0.575	0.574	0.578	0.688	0.558
	10d	0.581	0.589	0.644	0.511	0.551
	20d	0.643	0.714	0.692	0.625	0.628
	65d	0.759	0.789	0.782	0.778	0.748
	250d	0.703	0.801	0.803	0.793	0.817
AMZN	1d	0.552	0.465	0.321	0.455	0.359
	5d	0.565	0.374	0.443	0.429	0.553
	10d	0.545	0.368	0.257	0.334	0.331
	20d	0.611	0.570	0.509	0.379	0.494
	65d	0.670	0.683	0.704	0.657	0.715
	250d	0.993	0.986	0.987	0.984	0.984
CSCO	1d	0.537	0.428	0.486	0.485	0.449
	5d	0.574	0.546	0.561	0.593	0.529
	10d	0.600	0.540	0.525	0.482	0.577
	20d	0.602	0.473	0.616	0.559	0.568
	65d	0.710	0.606	0.682	0.656	0.682
	250d	0.822	0.801	0.836	0.778	0.805
GE	1d	0.532	0.525	0.473	0.433	0.461
	5d	0.515	0.566	0.540	0.533	0.584
	10d	0.498	0.514	0.551	0.524	0.602
	20d	0.544	0.613	0.656	0.579	0.656
	65d	0.470	0.534	0.580	0.518	0.527
	250d	0.623	0.678	0.683	0.625	0.771
GOOGL	1d	0.525	0.471	0.494	0.546	0.526
	5d	0.509	0.536	0.502	0.518	0.456
	10d	0.523	0.453	0.452	0.357	0.438
	20d	0.520	0.477	0.470	0.407	0.491
	65d	0.588	0.586	0.588	0.583	0.642
	250d	0.829	0.925	0.912	0.810	0.909

F-Maße ohne Feature Extracation	Horizont	Dummy	DT	RF	TunedDT	TunedRF
HP	1d	0.491	0.486	0.466	0.471	0.504
	5d	0.472	0.477	0.439	0.483	0.420
	10d	0.457	0.436	0.361	0.420	0.430
	20d	0.405	0.423	0.429	0.479	0.506
	65d	0.349	0.574	0.672	0.505	0.635
	250d	0.205	0.418	0.637	0.657	0.647
IBM	1d	0.489	0.553	0.605	0.605	0.555
	5d	0.532	0.541	0.496	0.498	0.577
	10d	0.513	0.501	0.451	0.492	0.495
	20d	0.425	0.469	0.334	0.468	0.441
	65d	0.419	0.582	0.524	0.576	0.495
	250d	0.266	0.677	0.796	0.820	0.853
ITNC	1d	0.573	0.525	0.545	0.503	0.501
	5d	0.589	0.559	0.556	0.553	0.602
	10d	0.601	0.529	0.569	0.506	0.577
	20d	0.609	0.546	0.556	0.528	0.587
	65d	0.602	0.661	0.690	0.636	0.618
	250d	0.619	0.573	0.697	0.595	0.696
MSFT	1d	0.509	0.371	0.333	0.297	0.298
	5d	0.577	0.440	0.412	0.328	0.472
	10d	0.603	0.436	0.479	0.471	0.386
	20d	0.634	0.648	0.586	0.735	0.450
	65d	0.736	0.698	0.731	0.697	0.655
	250d	0.927	0.842	0.901	0.892	0.958
WU	1d	0.526	0.420	0.419	0.332	0.355
	5d	0.593	0.507	0.503	0.514	0.493
	10d	0.557	0.472	0.439	0.409	0.462
	20d	0.536	0.555	0.531	0.530	0.547
	65d	0.624	0.607	0.634	0.617	0.576
	250d	0.688	0.752	0.781	0.739	0.828
XRX	1d	0.536	0.511	0.542	0.567	0.545
	5d	0.550	0.574	0.565	0.647	0.661
	10d	0.574	0.649	0.628	0.696	0.670
	20d	0.568	0.461	0.617	0.618	0.672
	65d	0.476	0.518	0.545	0.531	0.530
	250d	0.048	0.070	0.079	-1.000	0.009

A.7 Einfluss von Feature Extraction pro Aktie

Tabelle A.14 Einfluss von Feature Extraction pro Aktie

Datenset	DT	RF	TunedDT	TunedRF
AAPL	-0.011	-0.035	-0.020	0.004
AMZN	0.022	0.087	0.093	0.035
CSCO	0.026	-0.009	-0.027	-0.026
GE	-0.011	0.054	0.092	0.028
GOOGL	0.053	0.068	0.052	0.081
HP	0.014	0.031	0.032	-0.013
IBM	0.023	0.019	-0.066	-0.027
ITNC	-0.031	-0.032	0.026	-0.001
MSFT	0.050	0.062	0.052	0.074
WU	-0.035	0.005	0.017	-0.002
XRX	-0.019	-0.043	0.099	-0.021

A.8 Auswertungen der Klassifikatoren pro Gruppe

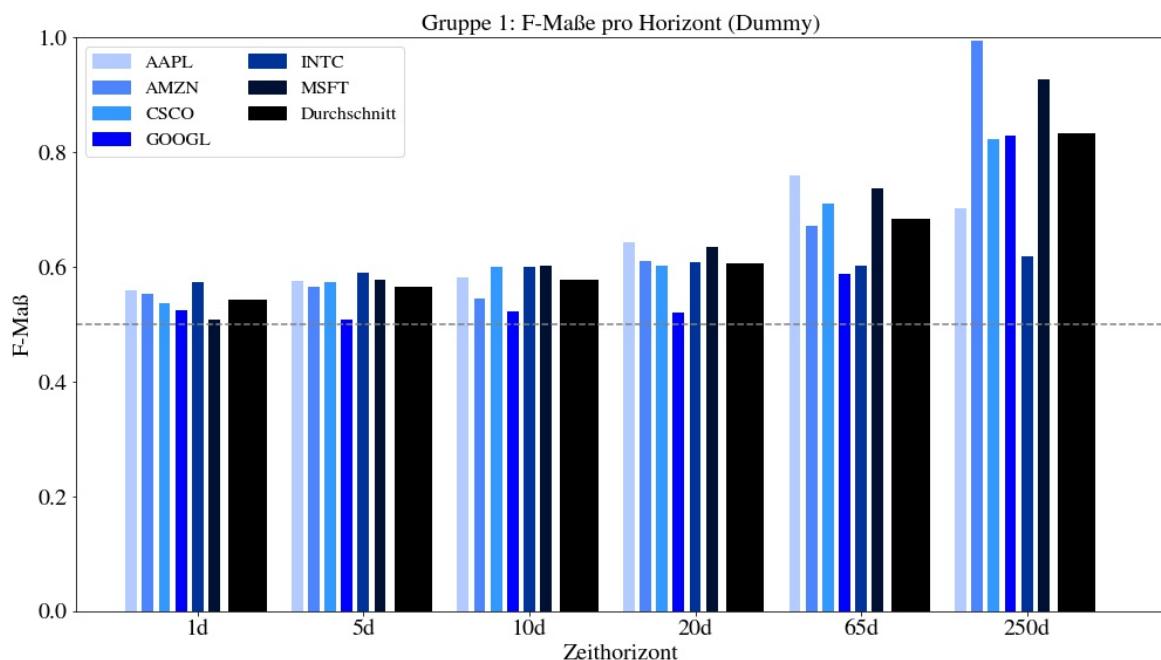


Abbildung A.28 Gruppe 1: F-Maße des Dummy Klassifikators pro Horizont pro Aktie

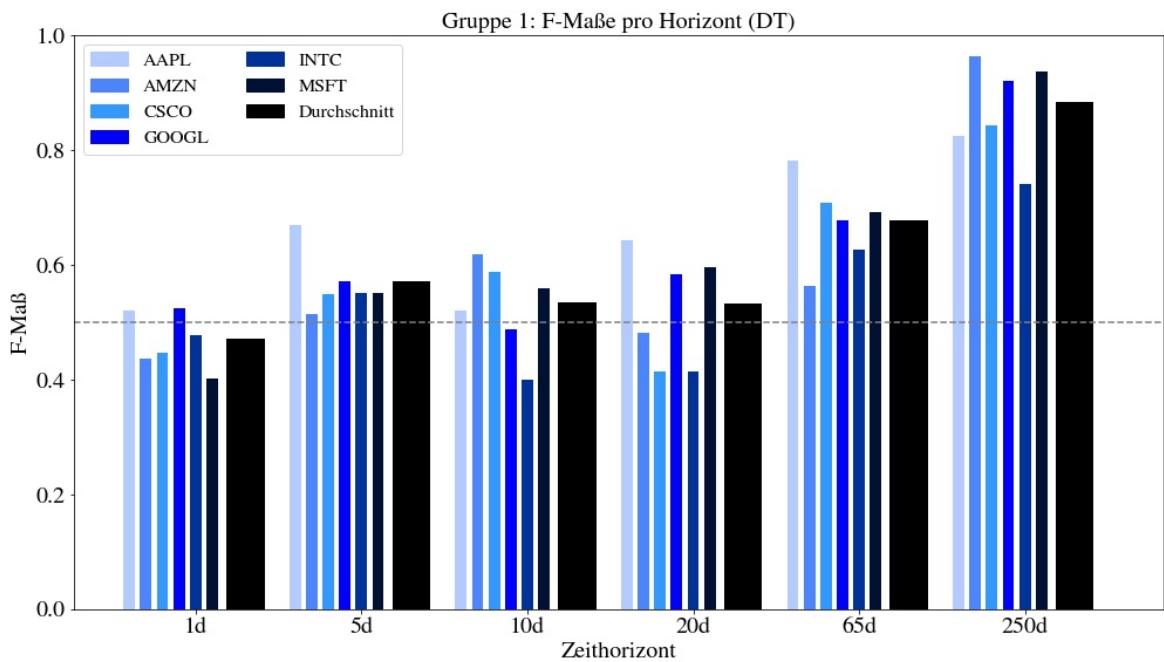


Abbildung A.29 Gruppe 1: F-Maße des Decision Trees pro Horizont pro Aktie

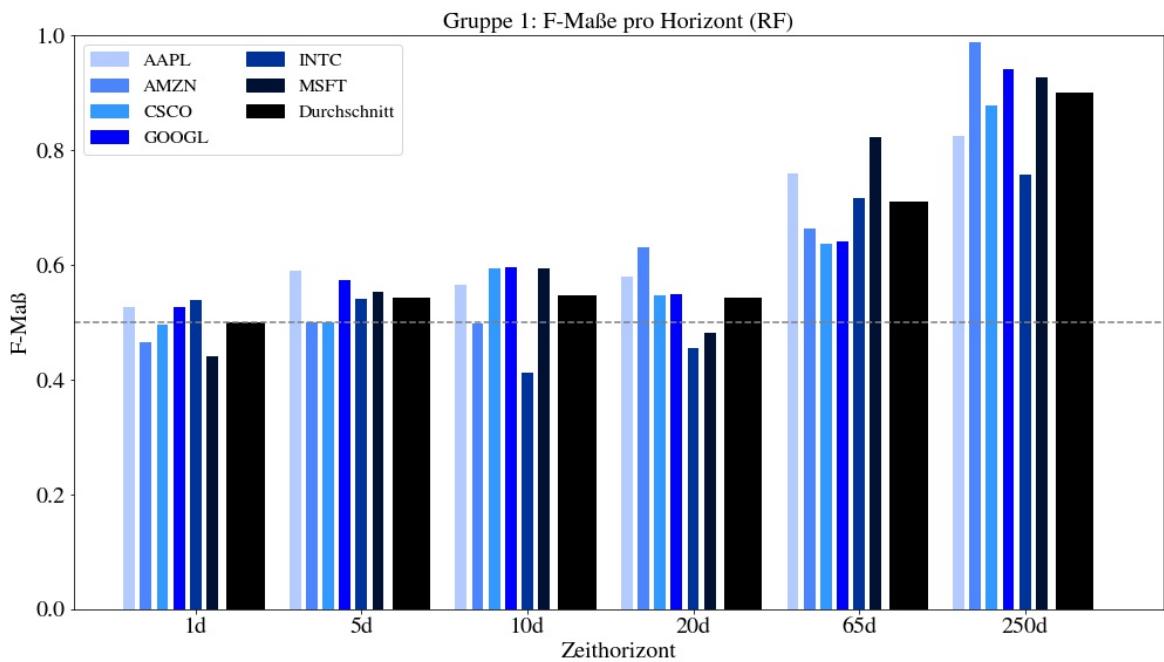


Abbildung A.30 Gruppe 1: F-Maße des Random Forests pro Horizont pro Aktie

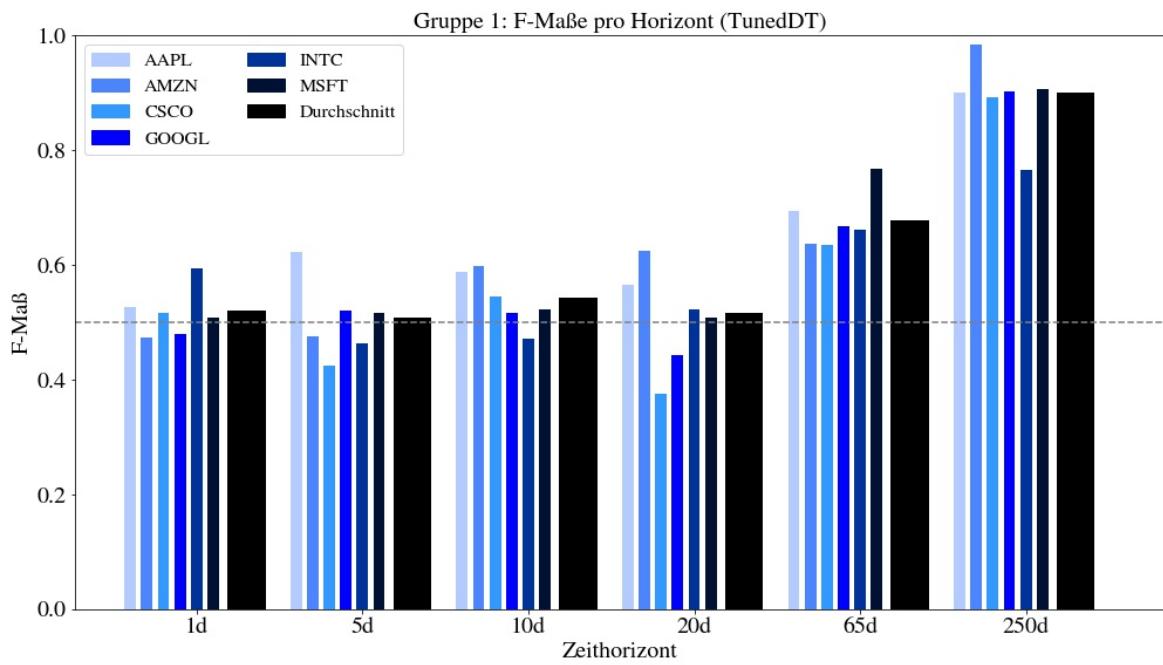


Abbildung A.31 Gruppe 1: F-Maße des Decision Trees mit Tuning pro Horizont pro Aktie

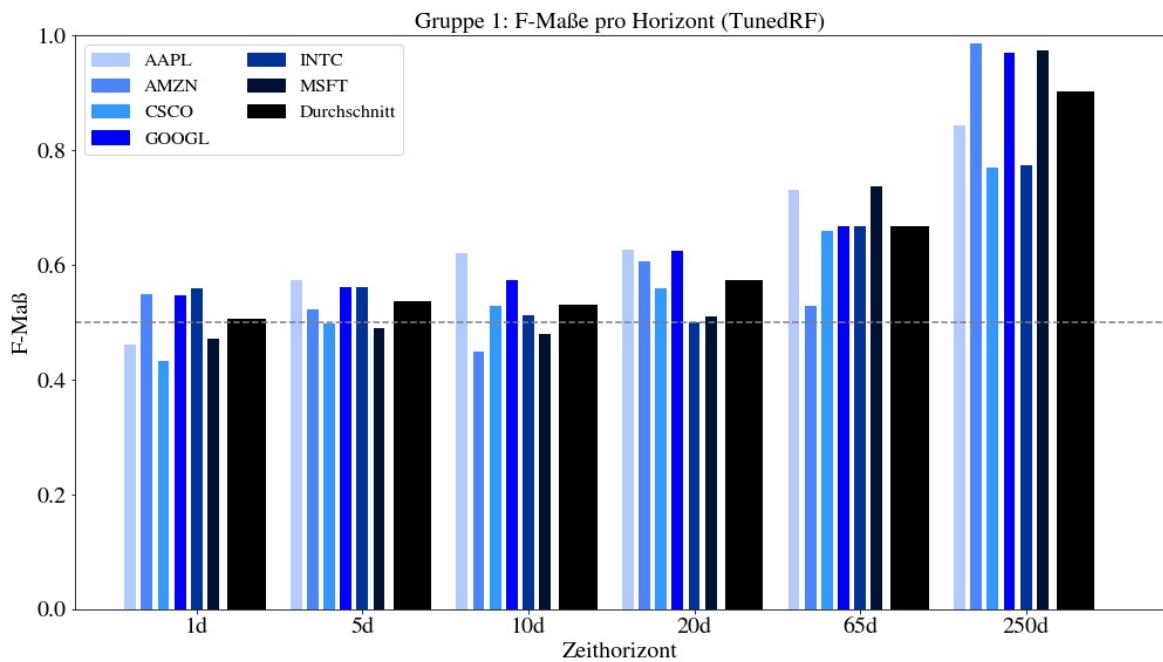


Abbildung A.32 Gruppe 1: F-Maße des Random Forests mit Tuning pro Horizont pro Aktie

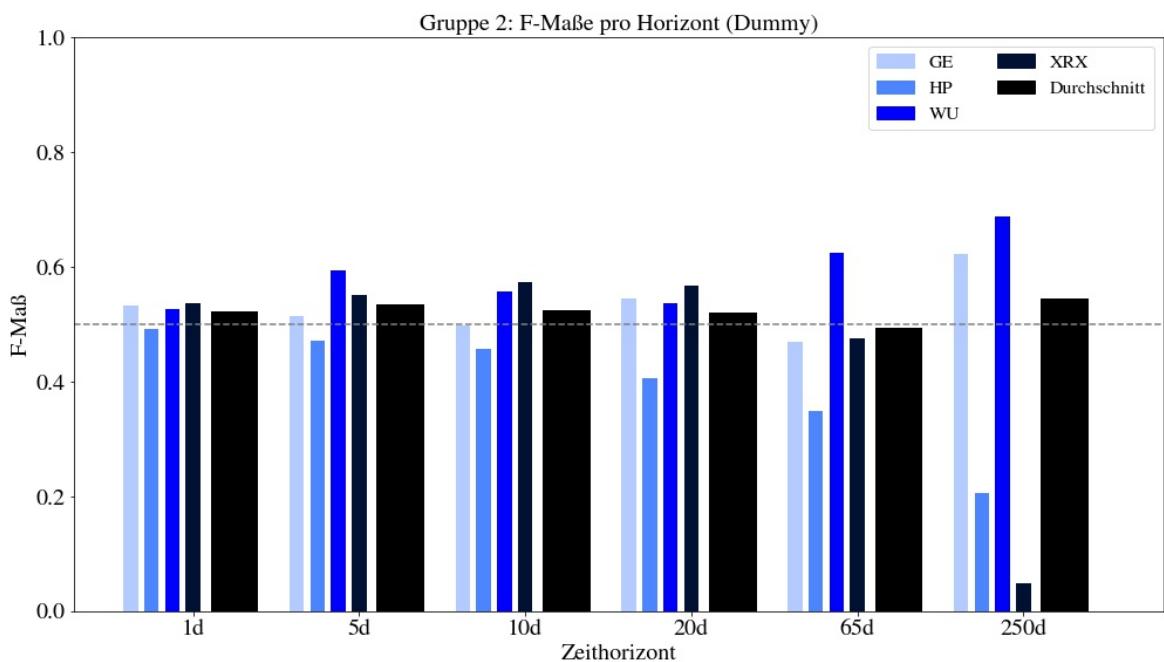


Abbildung A.33 Gruppe 2: F-Maße des Dummy Klassifikators pro Horizont pro Aktie

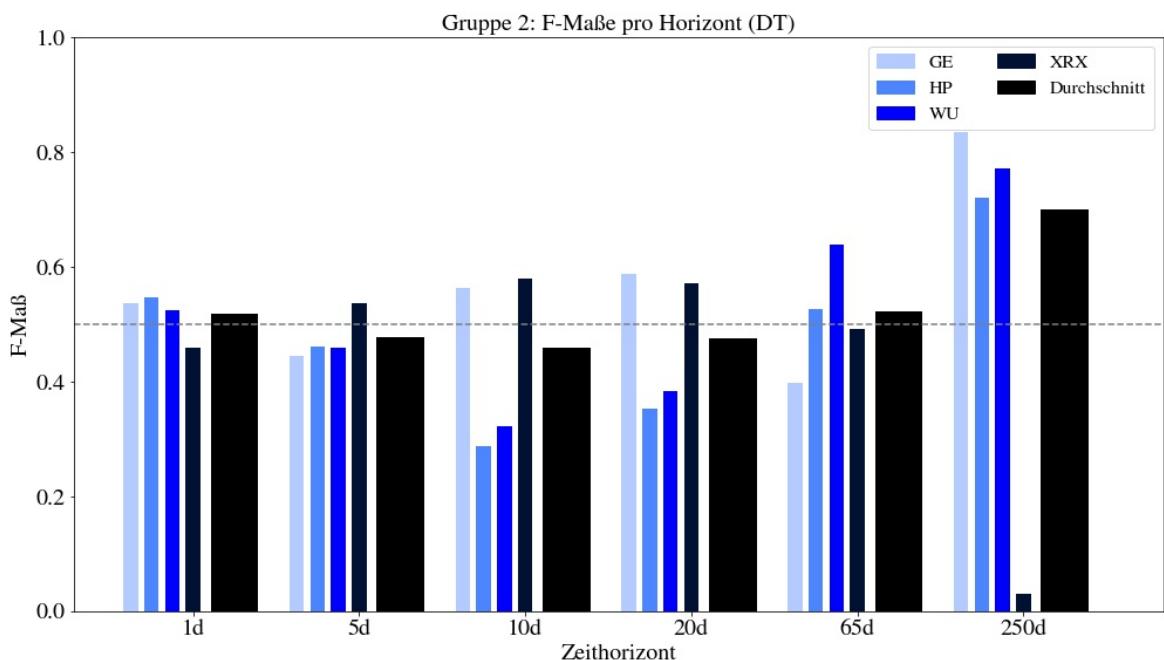


Abbildung A.34 Gruppe 2: F-Maße des Decision Trees pro Horizont pro Aktie

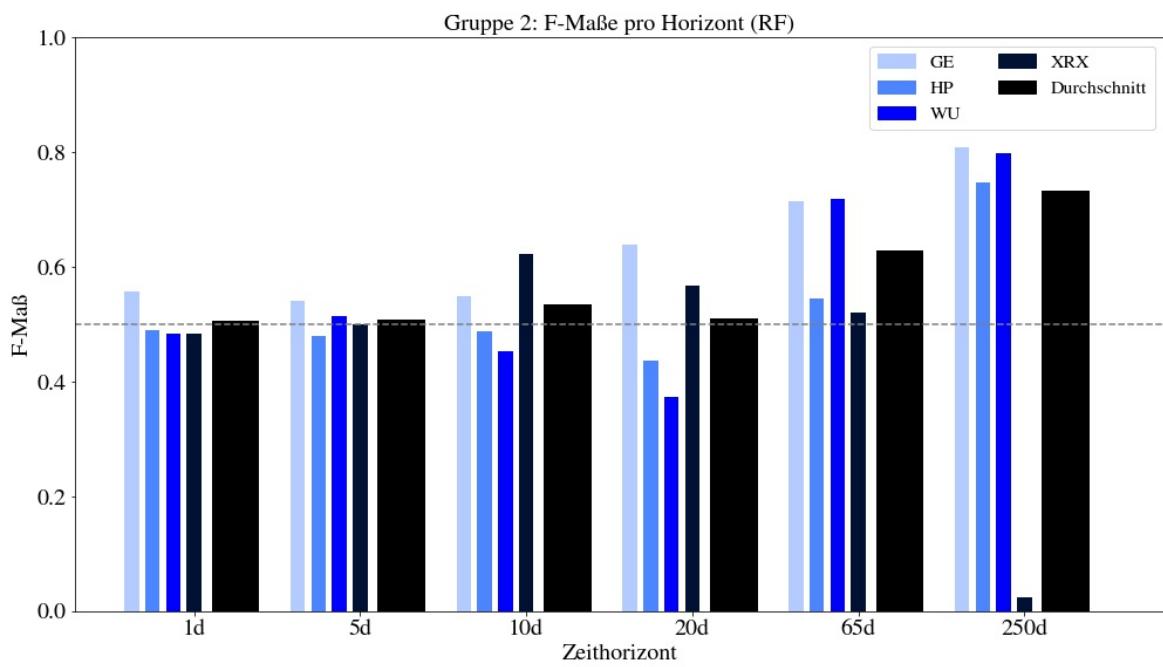


Abbildung A.35 Gruppe 2: F-Maße des Random Forests pro Horizont pro Aktie

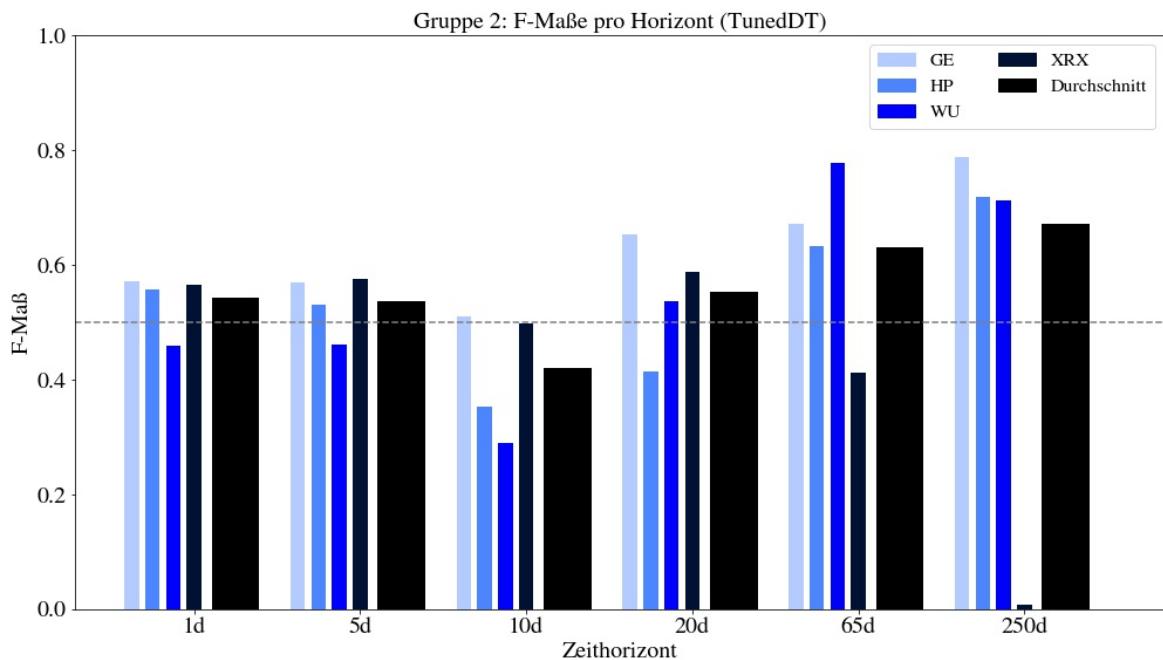


Abbildung A.36 Gruppe 2: F-Maße des Decision Trees mit Tuning pro Horizont pro Aktie

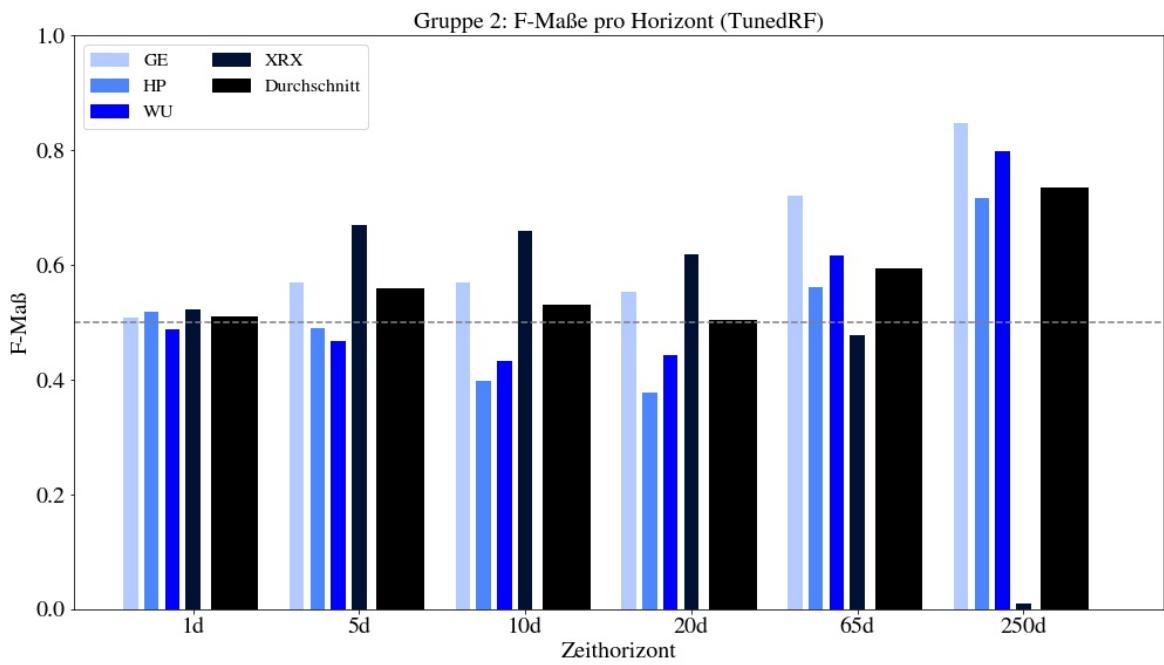


Abbildung A.37 Gruppe 2: F-Maße des Random Forests mit Tuning pro Horizont pro Aktie

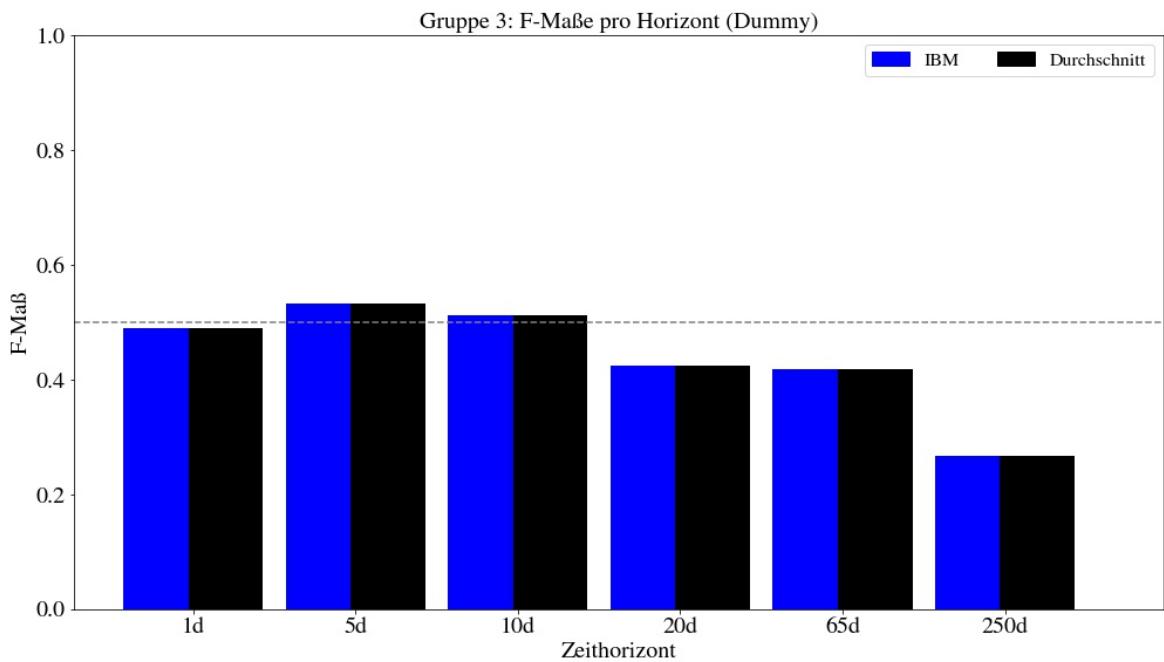


Abbildung A.38 Gruppe 3: F-Maße des Dummy Klassifikators pro Horizont pro Aktie

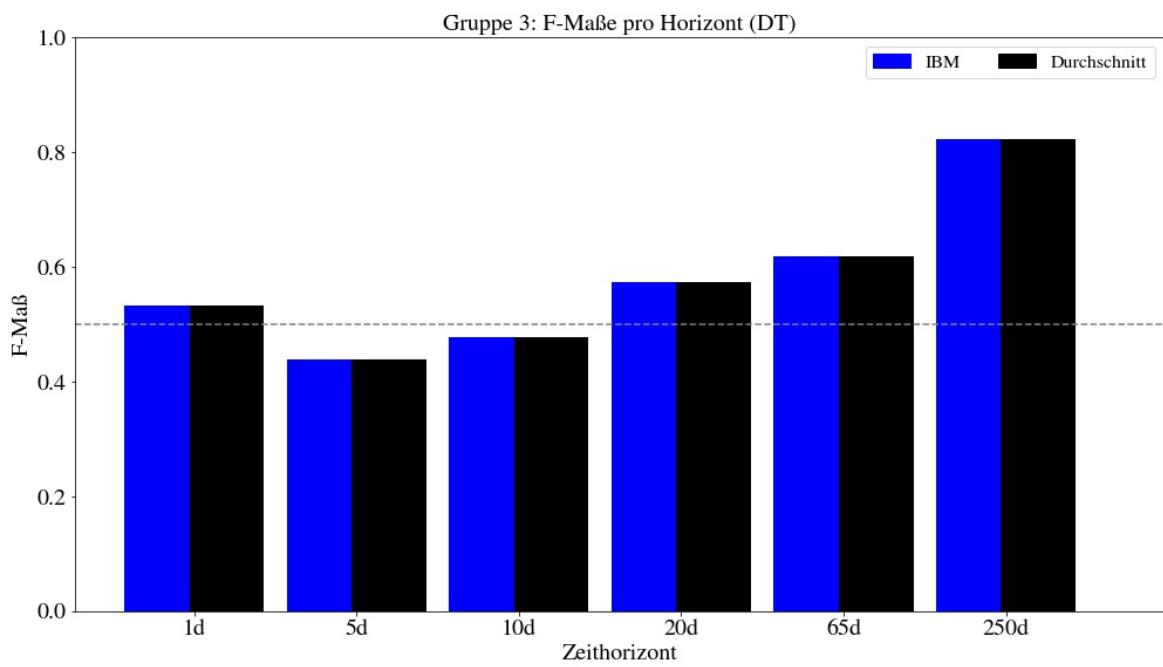


Abbildung A.39 Gruppe 3: F-Maße des Decision Trees pro Horizont pro Aktie

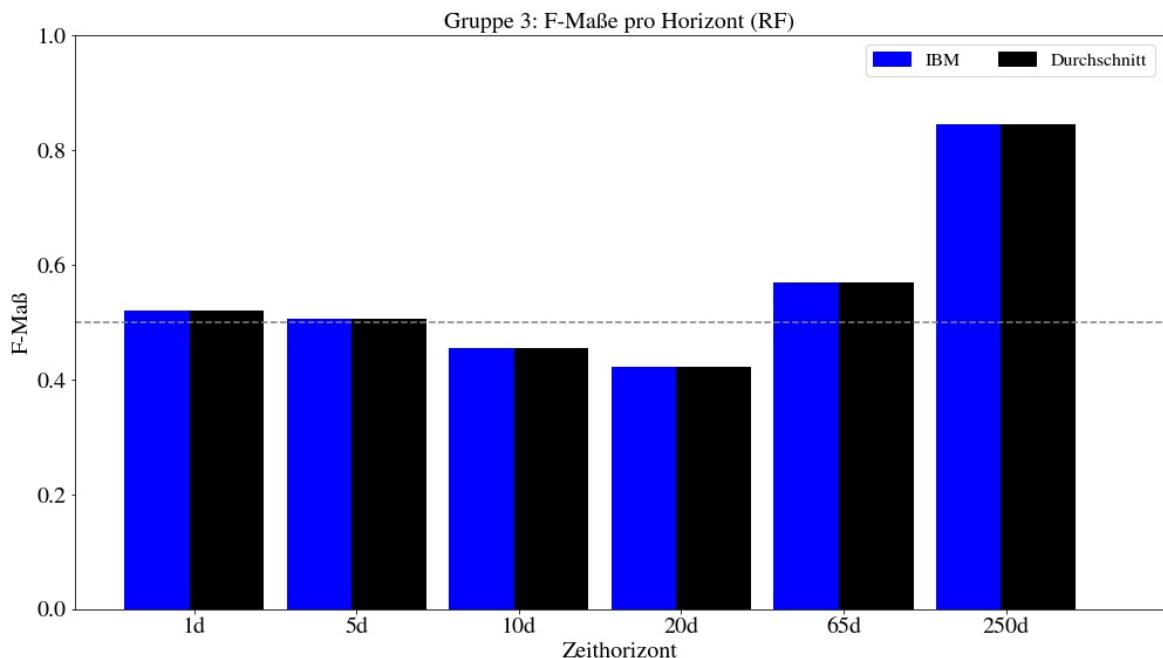


Abbildung A.40 Gruppe 3: F-Maße des Random Forests pro Horizont pro Aktie

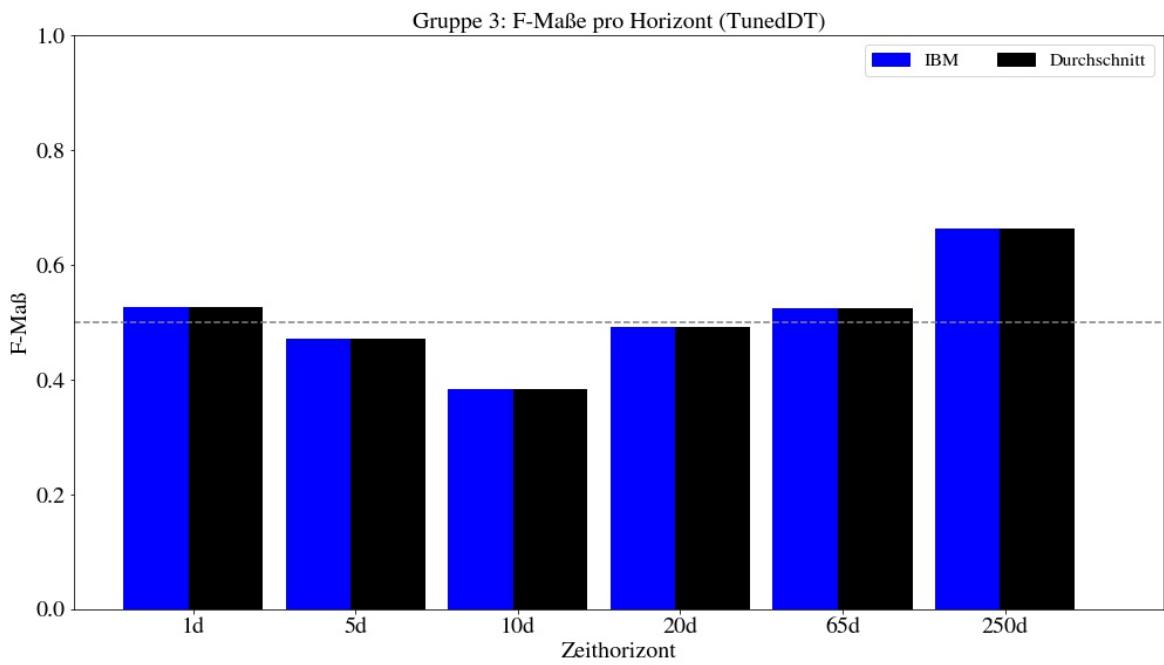


Abbildung A.41 Gruppe 3: F-Maße des Decision Trees mit Tuning pro Horizont pro Aktie

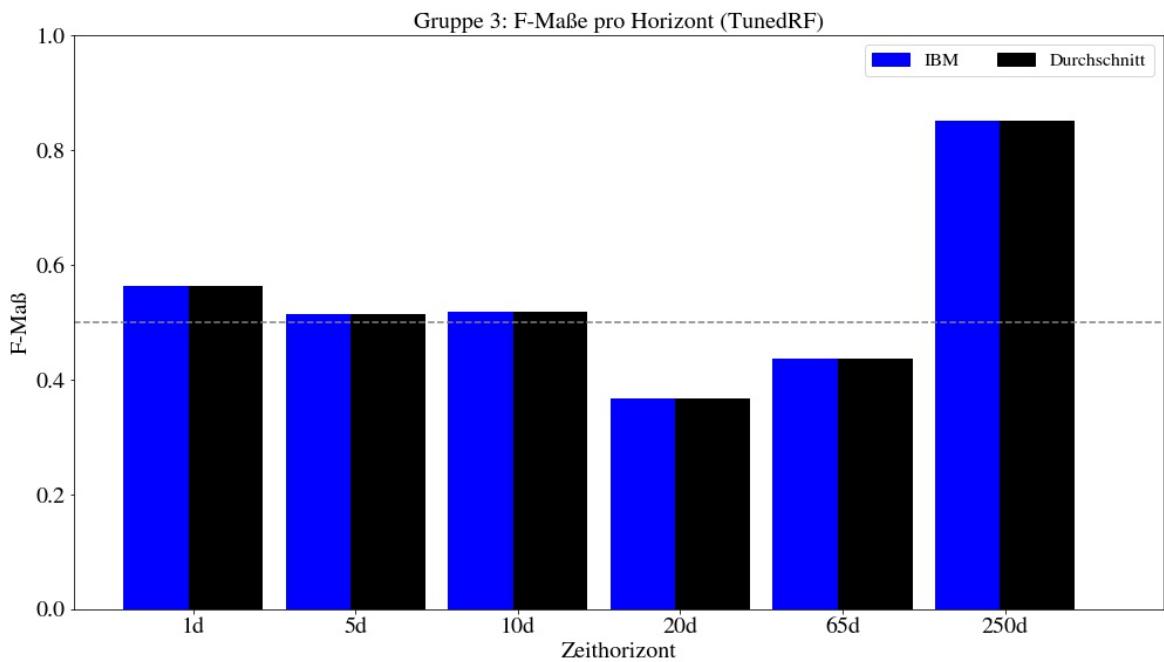


Abbildung A.42 Gruppe 3: F-Maße des Random Forests mit Tuning pro Horizont pro Aktie

A.9 Veränderung des F-Maßes durch Hyperparameter-Tuning

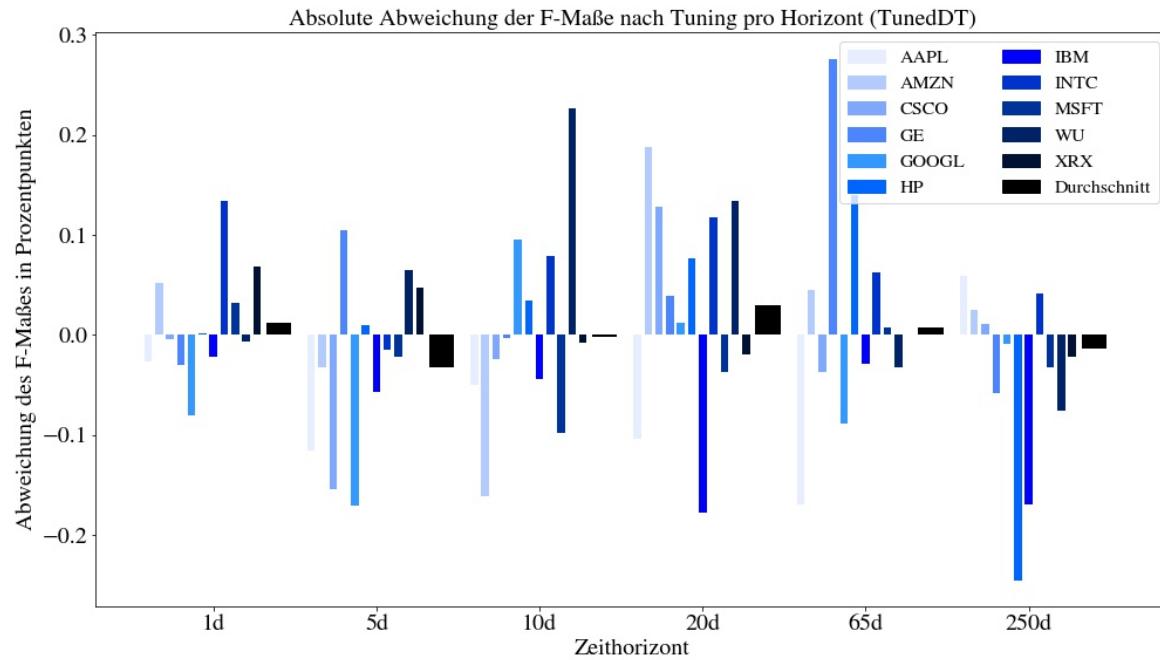


Abbildung A.43 Einfluss von 5TSCV Hyperparameter-Tuning auf Decision Trees

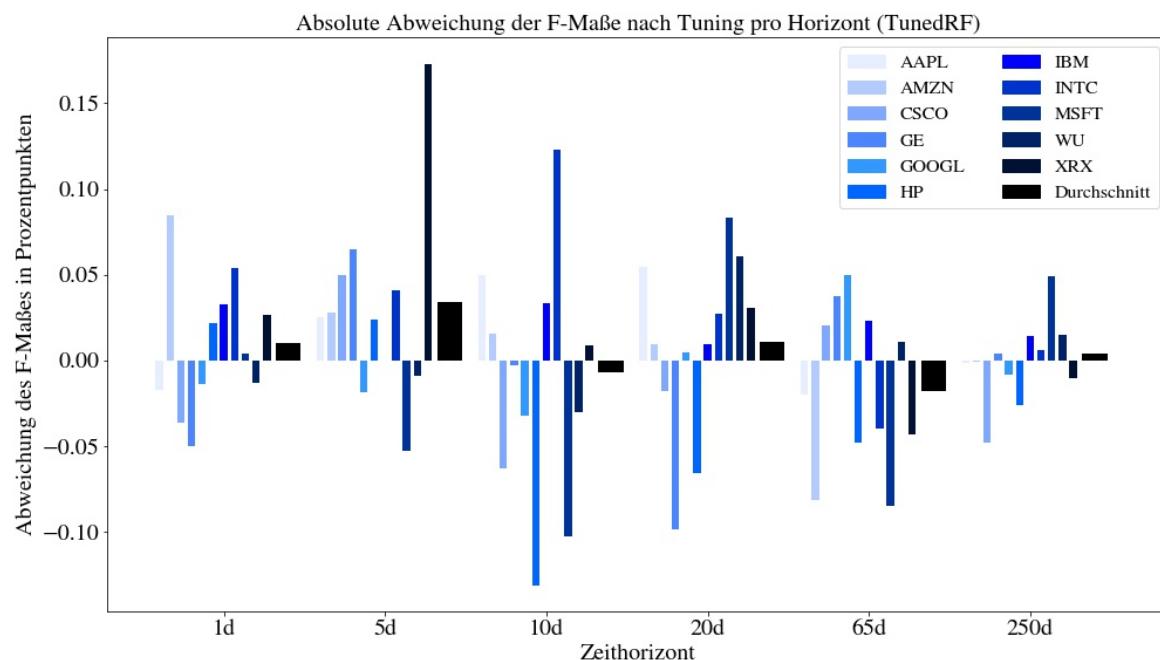


Abbildung A.44 Einfluss von 5TSCV Hyperparameter-Tuning auf Random Forests

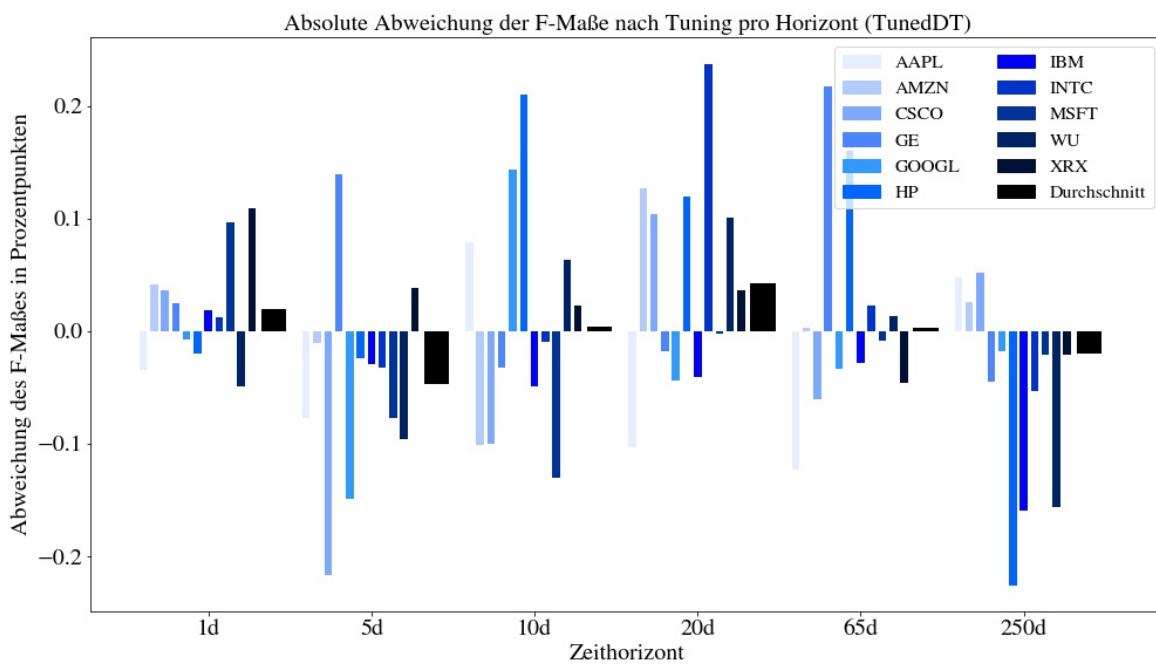


Abbildung A.45 Einfluss von 3TSCV Hyperparameter-Tuning auf Decision Trees

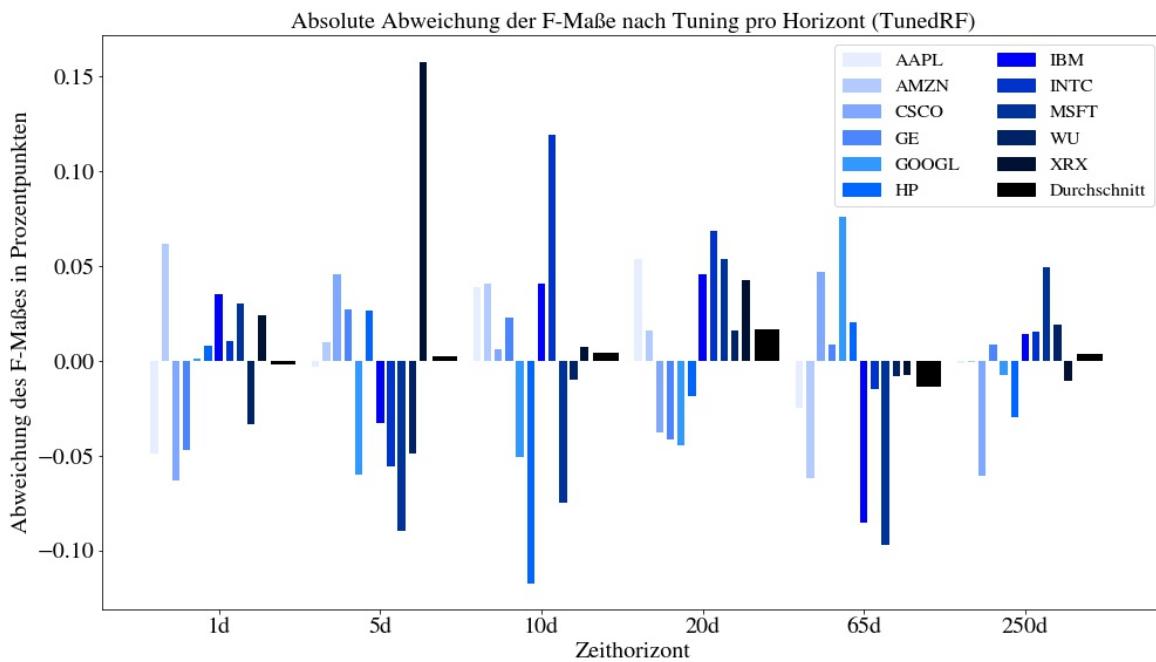


Abbildung A.46 Einfluss von 3TSCV Hyperparameter-Tuning auf Random Forests

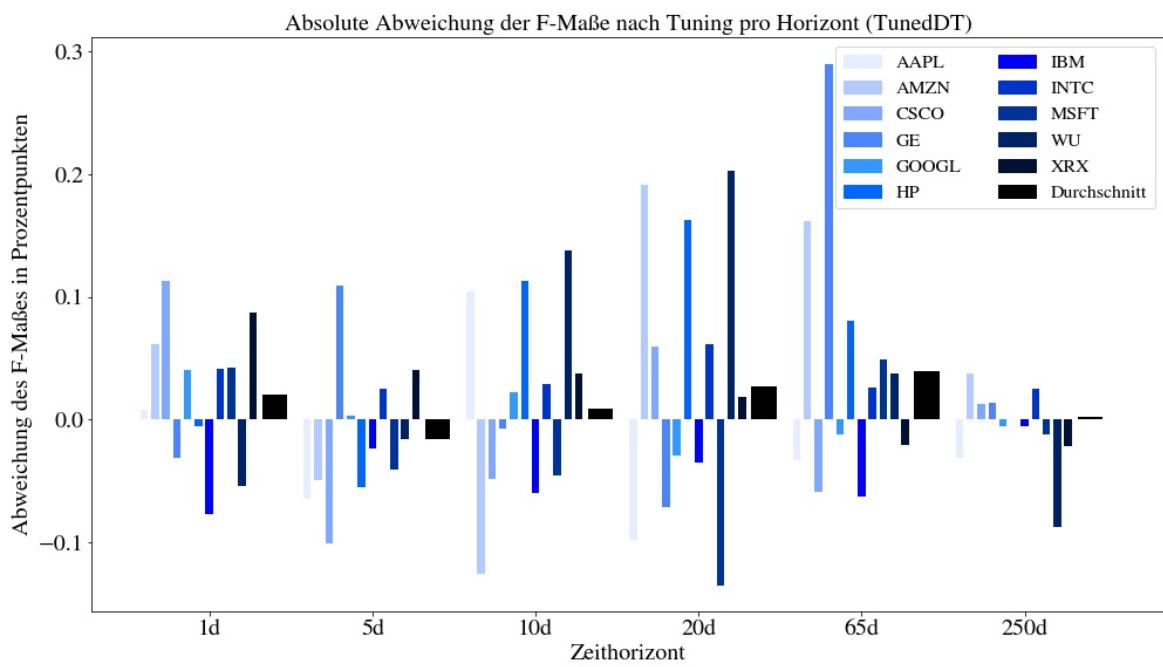


Abbildung A.47 Einfluss von 5fCV Hyperparameter-Tuning auf Decision Trees

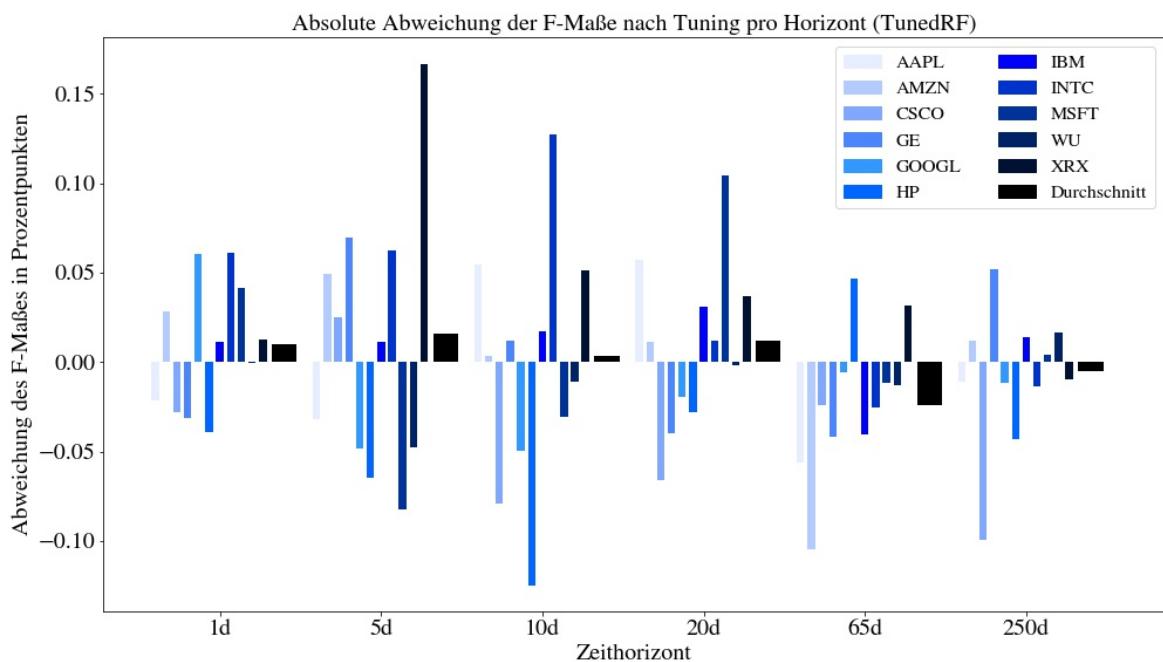


Abbildung A.48 Einfluss von 5fCV Hyperparameter-Tuning auf Random Forests

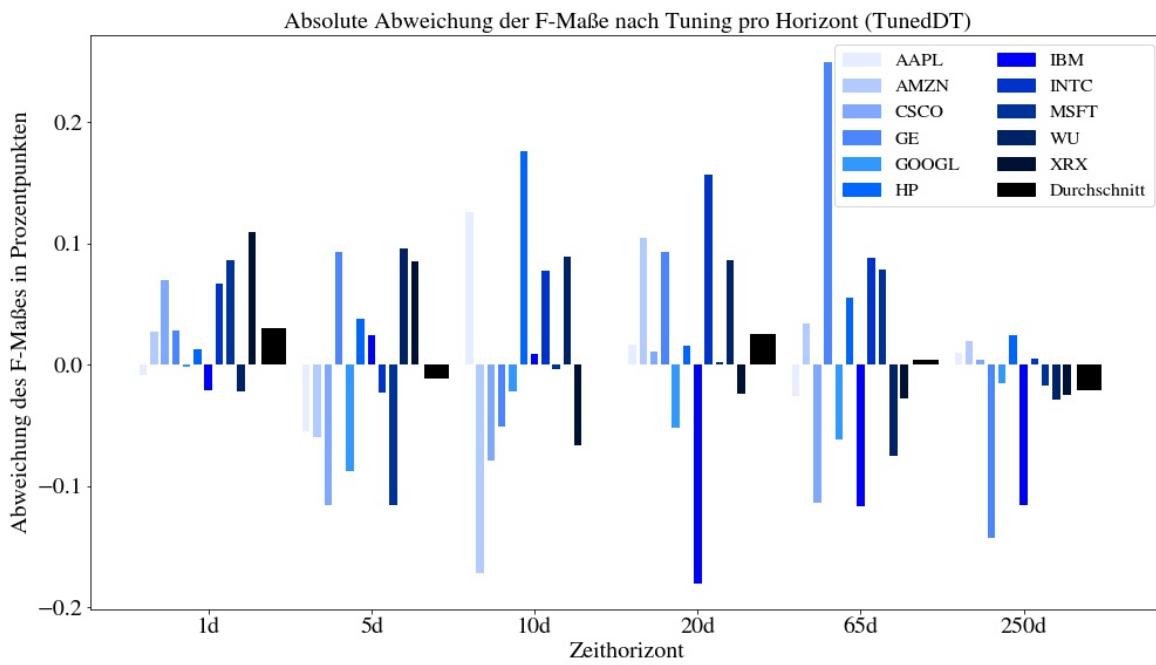


Abbildung A.49 Einfluss von 3fCV Hyperparameter-Tuning auf Decision Trees

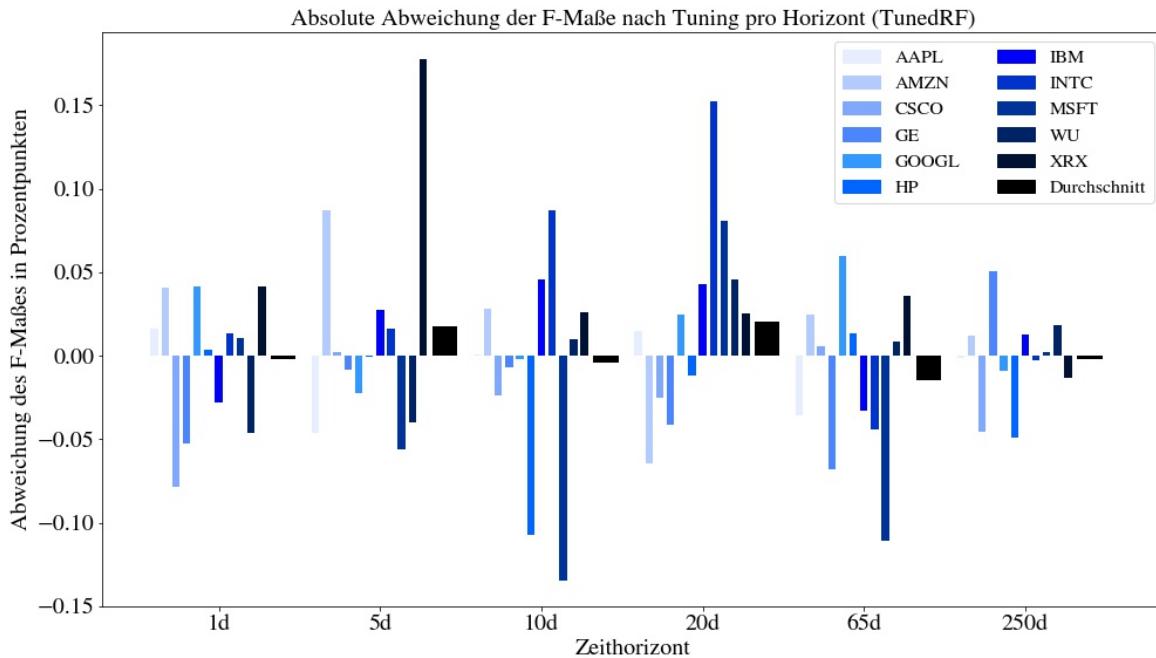


Abbildung A.50 Einfluss von 3fCV Hyperparameter-Tuning auf Random Forests

Literaturverzeichnis

[Abdou u. Pointon 2011] ABDOU, Hussein ; POINTON, John: Credit Scoring, Statistical Techniques and Evaluation Criteria: A Review of the Literature. In: *Int. Syst. in Accounting, Finance and Management* 18 (2011), 04, S. 59–88

[Alavi u. a. 2015] ALAVI, Seyed E. ; SINAEI, Hasanali ; AFSHARIRAD, Elham: Predict the trend of stock prices using machine learning techniques. In: *International Academic Journal of Economics* 2 (2015), S. 1–11

[Alpaydin 2008] ALPAYDIN, E.: *Maschinelles Lernen*. Oldenbourg, 2008. – ISBN 9783486581140

[Anand u. a. 2010] ANAND, Ashish ; PUGALENTHI, Ganesan ; FOGEL, Gary ; SUGANTHAN, Ponnuthurai: An approach for classification of highly imbalanced data using weighting and undersampling. In: *Amino acids* 39 (2010), 11, S. 1385–91. <http://dx.doi.org/10.1007/s00726-010-0595-2>. – DOI 10.1007/s00726-010-0595-2

[Belgiu u. Drăguț 2016] BELGIU, Mariana ; DRĂGUȚ, Lucian: Random forest in remote sensing: A review of applications and future directions. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 114 (2016), 04, S. 24–31

[Bergmeir u. Benitez 2011] BERGMEIR, C. ; BENITEZ, J. M.: Forecaster performance evaluation with cross-validation and variants. In: *2011 11th International Conference on Intelligent Systems Design and Applications*, 2011, S. 849–854

[Biau u. Scornet 2016] BIAU, Gérard ; SCORNET, Erwan: A random forest guided tour. In: *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research* 25 (2016), Nr. 2, S. 197–227

[BlackRock Inc. 2020a] Aladdin - Benefits to risk managers. In: *BlackRock Inc.* Webseite (2020). <https://www.blackrock.com/aladdin/benefits/risk-managers>. – Besucht: 02. Januar 2020

[BlackRock Inc. 2020b] Risikomanagement mit Aladdin. In: *BlackRock Inc.* Webseite (2020). <https://www.blackrock.com/at/finanzberater-und-banken/uber-blackrock/risk-management-with-aladdin>. – Besucht: 02. Januar 2020

[BMBF 2019] Wissenschaftsjahr 2019-Künstliche Intelligenz offiziell gestartet. In: *Bundesministerium für Bildung und Forschung* Webseite (2019). <https://www.bildung-forschung.digital/de/wissenschaftsjahr-2019---kuenstliche-intelligenz-offiziell-gestartet-2507.html>. – Besucht: 02. Januar 2020

[Breiman 2001] BREIMAN, Leo: Random Forests. In: *Machine Learning* 45 (2001), Oktober, Nr. 1, S. 5–32. <http://dx.doi.org/10.1023/A:1010933404324>. – DOI 10.1023/A:1010933404324. – ISSN 0885–6125

- [Brown u. Sandholm 2019] BROWN, Noam ; SANDHOLM, Tuomas: Superhuman AI for multiplayer poker. In: *Science* 365 (2019), Nr. 6456, S. 885–890. <http://dx.doi.org/10.1126/science.aay2400>. – DOI 10.1126/science.aay2400
- [Campbell u. a. 2002] CAMPBELL, Murray ; HOANE, A.Joseph ; HSU, Feng hsiung: Deep Blue. In: *Artificial Intelligence* 134 (2002), Nr. 1, S. 57 – 83. [http://dx.doi.org/10.1016/S0004-3702\(01\)00129-1](http://dx.doi.org/10.1016/S0004-3702(01)00129-1). – DOI 10.1016/S0004-3702(01)00129-1. – ISSN 0004-3702
- [Cerqueira u. a. 2019] CERQUEIRA, Vítor ; TORGÓ, Luís ; MOZETIC, Igor: Evaluating time series forecasting models: An empirical study on performance estimation methods. In: *ArXiv abs/1905.11744* (2019)
- [Cui u. a. 2018] CUI, L. ; CHEN, J. ; WU, W.: Predicting Secondary Equity Offerings (SEOs) Using Machine Learning. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, S. 1219–1224
- [Cutler u. a. 2007] CUTLER, D. R. ; EDWARDS, Thomas C. ; BEARD, Karen H. ; CUTLER, Adele ; HESS, Kyle ; GIBSON, Jacob ; LAWLER, Joshua J.: Random forests for classification in ecology. In: *Ecology* 88 11 (2007), S. 2783–92. <http://dx.doi.org/10.1890/07-0539.1>. – DOI 10.1890/07-0539.1
- [Di 2014] DI, Xinjie: Stock Trend Prediction with Technical Indicators using SVM / Stanford University. 2014. – Forschungsbericht
- [Diaz-Uriarte u. Alvarez 2006] DIAZ-URIARTE, Ramon ; ALVAREZ, Sara: Gene Selection and Classification of Microarray Data Using Random Forest. In: *BMC bioinformatics* 7 (2006), 02, S. 3. <http://dx.doi.org/10.1186/1471-2105-7-3>. – DOI 10.1186/1471-2105-7-3
- [Drakopoulou 2016] DRAKOPOULOU, Veliota: A Review of Fundamental and Technical Stock Analysis Techniques. In: *Journal of Stock & Forex Trading* 05 (2016), 01. <http://dx.doi.org/10.4172/2168-9458.1000163>. – DOI 10.4172/2168-9458.1000163
- [Economist 2019] The rise of the financial machines. In: *The Economist* (2019), Oct. <https://www.economist.com/leaders/2019/10/03/the-rise-of-the-financial-machines>. – Besucht: 02. Januar 2020
- [Fama 1965] FAMA, Eugene F.: The Behavior of Stock-Market Prices. In: *The Journal of Business* 38 (1965), Nr. 1, S. 34–105. <http://dx.doi.org/10.1086/294743>. – DOI 10.1086/294743
- [Feurer u. Hutter 2019] FEURER, Matthias ; HUTTER, Frank: *Automated Machine Learning - Methods, Systems, Challenges*. Springer International Publishing, 2019. <http://dx.doi.org/10.1007/978-3-030-05318-5>. – ISBN 978-3-030-05318-5
- [Freund u. Schapire 1997] FREUND, Yoav ; SCHAPIRE, Robert E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. In: *Journal of Computer and System Sciences* 55 (1997), Nr. 1, S. 119 – 139. <http://dx.doi.org/10.1006/jcss.1997.1504>. – DOI 10.1006/jcss.1997.1504
- [Gron 2017] GRON, Aurlien: *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st. O'Reilly Media, Inc., 2017. – ISBN 1491962291, 9781491962299

- [Gupta u. a. 2017] GUPTA, Bhumika ; RAWAT, Aditya ; JAIN, Akshay ; ARORA, Arpit ; DHAMI, Naresh: Analysis of Various Decision Tree Algorithms for Classification in Data Mining. In: *International Journal of Computer Applications* 163 (2017), 04, S. 15–19. <http://dx.doi.org/10.5120/ijca2017913660>. – DOI 10.5120/ijca2017913660
- [Haibo He u. a. 2008] HAIBO HE ; YANG BAI ; GARCIA, E. A. ; SHUTAO LI: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, S. 1322–1328
- [Harding u. Barden 1997] HARDING, Luke ; BARDEM, Leonard: Deep Blue win a giant step for computerkind. In: *The Guardian* (1997). <https://www.theguardian.com/theguardian/2011/may/12/deep-blue-beats-kasparov-1997>. – Besucht: 02. Januar 2020
- [Johnson 1997] JOHNSON, George: Deep, Deeper, Deepest Blue. In: *The New York Times* (1997). <https://www.nytimes.com/1997/05/18/weekinreview/deep-deeper-deepest-blue.html>. – Besucht: 02. Januar 2020
- [Karliczek 2019] KARLICZEK, Anja: Rede der Bundesministerin für Bildung und Forschung, Anja Karliczek, zur Strategie Künstliche Intelligenz vor dem Deutschen Bundestag am 15. Februar 2019 in Berlin. In: *Bulletin 20-1 der Bundesregierung* (2019). <https://www.bundesregierung.de/breg-de/service/bulletin/rede-der-bundesministerin-fuer-bildung-und-forschung-anja-karliczek-1581192>. – Besucht: 02. Januar 2020
- [Khaidem u. a. 2016] KHAIDEM, Luckyson ; SAHA, Snehanshu ; DEY, Sudeepa R.: Predicting the direction of stock market prices using random forest. In: *ArXiv abs/1605.00003* (2016)
- [Lendasse u. a. 2000] LENDASSE, Amaury ; BODT, E. ; WERTZ, Vincent ; VERLEYSEN, Michel: Non-linear financial time series forecasting - Application to the Bel 20 stock market index. In: *European Journal of Economic and Social Systems* 14 (2000), 11, Nr. 1, S. 81–91. <http://dx.doi.org/10.1051/ejess:2000110>. – DOI 10.1051/ejess:2000110
- [Li 2016] LI, C.: The application of high-dimensional data classification by random forest based on hadoop cloud computing platform. 51 (2016), 01, S. 385–390. <http://dx.doi.org/10.3303/CET1651065>. – DOI 10.3303/CET1651065
- [Lohrmann u. Luukka 2019] LOHRMANN, Christoph ; LUUKKA, Pasi: Classification of intraday S&P500 returns with a Random Forest. In: *International Journal of Forecasting* 35 (2019), Nr. 1, S. 390–407. <http://dx.doi.org/10.1016/j.ijforecast.2018>. – DOI 10.1016/j.ijforecast.2018
- [Maini u. Govinda 2017] MAINI, S. S. ; GOVINDA, K.: Stock market prediction using data mining techniques. In: *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, 2017, S. 654–661
- [McCarthy u. a. 1955] McCARTHY, J. ; MINSKY, M. L. ; ROCHESTER, N. ; SHANNON, C. E.: *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. 1955
- [Mitchell 1997] MITCHELL, Thomas M.: *Machine Learning*. 1. New York, NY, USA : McGraw-Hill, Inc., 1997. – ISBN 0070428077, 9780070428072

[Pasupulety u. a. 2019] PASUPULETY, U. ; ABDULLAH ANEES, A. ; ANMOL, S. ; MOHAN, B. R.: Predicting Stock Prices using Ensemble Learning and Sentiment Analysis. In: *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2019, S. 215–222

[Pedregosa u. a. 2011] PEDREGOSA, Fabian ; VAROQUAUX, Gaël ; GRAMFORT, Alexandre ; MICHEL, Vincent ; THIRION, Bertrand ; GRISEL, Olivier ; BLONDEL, Mathieu ; PRETTENHOFER, Peter ; WEISS, Ron ; DUBOURG, Vincent ; VANDERPLAS, Jake ; PASSOS, Alexandre ; COURNAPEAU, David ; BRUCHER, Matthieu ; PERROT, Matthieu ; DUCHESNAY, Édouard: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), November, S. 2825–2830

[Powers 2014] POWERS, David: What the F-measure doesn't measure / Beijing University of Technology, China and Flinders University, Australia. 2014. – Forschungsbericht

[Pozen u. Ruane 2019] POZEN, Robert C. ; RUANE, Jonathan: What Machine Learning Will Mean for Asset Managers. In: *Harvard Business Review* (2019)

[Prasad u. a. 2006] PRASAD, Anantha ; IVERSON, Louis ; LIAW, Andy: Newer Classification and Regression Tree Techniques: Bagging and Random Forests for Ecological Prediction. In: *Ecosystems* 9 (2006), 03, S. 181–199. <http://dx.doi.org/10.1007/s10021-005-0054-1>. – DOI 10.1007/s10021-005-0054-1

[Probst u. a. 2019] PROBST, Philipp ; WRIGHT, Marvin N. ; BOULESTEIX, Anne-Laure: Hyperparameters and tuning strategies for random forest. In: *WIREs Data Mining and Knowledge Discovery* 9 (2019), Nr. 3. <http://dx.doi.org/10.1002/widm.1301>. – DOI 10.1002/widm.1301

[Quinlan 1986] QUINLAN, J. R.: Induction of decision trees. In: *Machine Learning* 1 (1986), Mar, Nr. 1, S. 81–106. <http://dx.doi.org/10.1007/BF00116251>. – DOI 10.1007/BF00116251

[Russell u. Norvig 2009] RUSSELL, Stuart ; NORVIG, Peter: *Artificial Intelligence: A Modern Approach*. 3rd. Upper Saddle River, NJ, USA : Prentice Hall Press, 2009. – ISBN 0136042597, 9780136042594

[Sadia u. a. 2019] SADIA, K. H. ; SHARMA, Aditya ; PAUL, Adarrsh ; PADHI, Sarmistha ; SANYAL, Saurav: Stock Market Prediction Using Machine Learning Algorithms. In: *International Journal of Engineering and Advanced Technology* 8 (2019), 04

[Shannon 1950] SHANNON, Claude E.: Programming a Computer for Playing Chess. In: *Philosophical Magazine* 41 (1950), Nr. 314, S. 256–275. <http://dx.doi.org/10.1080/14786445008521796>. – DOI 10.1080/14786445008521796

[Shannon 1948] SHANNON, Claude E.: A Mathematical Theory of Communication. In: *The Bell System Technical Journal* 27 (1948), 7, Nr. 3, S. 379–423. <http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x>. – DOI 10.1002/j.1538-7305.1948.tb01338.x

[Shoham u. a. 2018] SHOHAM, Yoav ; PERRAULT, Raymond ; BRYNJOLFSSON, Erik u. a.: The AI Index 2018 Annual Report / AI Index Steering Committee, Human-Centered AI Initiative, Stanford University, Stanford, CA. 2018. – Forschungsbericht

- [Shotton u. a. 2011] SHOTTON, J. ; FITZGIBBON, A. ; COOK, M. ; SHARP, T. ; FINOCCHIO, M. ; MOORE, R. ; KIPMAN, A. ; BLAKE, A.: Real-time Human Pose Recognition in Parts from Single Depth Images. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2011 (CVPR '11), S. 1297–1304
- [Silver u. Hassabis 2016] SILVER, David ; HASSABIS, Demis: AlphaGo: Mastering the ancient game of Go with Machine Learning. In: *Google AI Blog* (2016), Jan. <https://ai.googleblog.com/2016/01/alphago-mastering-ancient-game-of-go.html>. – Besucht: 02. Januar 2020
- [Silver u. a. 2017] SILVER, David ; SCHRITTWIESER, Julian ; SIMONYAN, Karen ; ANTONOGLOU, Ioannis ; HUANG, Aja ; GUEZ, Arthur ; HUBERT, Thomas ; BAKER, L R. ; LAI, Matthew ; BOLTON, Adrian ; CHEN, Yutian ; LILLICRAP, Timothy P. ; HU, Fan Fong C. ; SIFRE, Laurent ; DRIESSCHE, George van d. ; GRAEPEL, Thore ; HASSABIS, Demis: Mastering the game of Go without human knowledge. In: *Nature* 550 (2017), S. 354–359
- [Svetnik u. a. 2003] SVETNIK, Vladimir ; LIAW, Andy ; TONG, Christopher ; CULBERSON, J. C. ; SHERIDAN, Robert P. ; FEUSTON, Bradley P.: Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. In: *Journal of Chemical Information and Computer Sciences* 43 (2003), Nr. 6, S. 1947–1958
- [Szmigiera 2019] SZMIGIERA, M.: *Largest asset management companies worldwide as of March 2019, by managed assets (in trillion U.S. dollars)*. <https://www.statista.com/statistics/431790/leading-asset-management-companies-worldwide-by-assets>. Version: 2019. – Besucht: 02. Januar 2020
- [Tang u. a. 2018] TANG, Cheng ; GARREAU, Damien ; LUXBURG, Ulrike von: When do random forests fail? In: *Proceedings Neural Information Processing Systems*, 2018
- [Thomas 2013] THOMAS, Evan: *Ike's Bluff: President Eisenhower's Secret Battle to Save the World*. Back Bay Books, 2013. – ISBN 978–0316091039
- [Tin Kam Ho 1998] TIN KAM HO: The random subspace method for constructing decision forests. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1998), Aug, Nr. 8, S. 832–844. <http://dx.doi.org/10.1109/34.709601>. – DOI 10.1109/34.709601
- [Turney 2000] TURNEY, Peter: Types of Cost in Inductive Concept Learning. In: *17th International Conference on Machine Learning*, 2000
- [Verleysen u. François 2005] VERLEYSEN, Michel ; FRANÇOIS, Damien: The Curse of Dimensionality in Data Mining and Time Series Prediction, 2005, S. 758–770
- [Whitley u. Watson 2005] KAPITEL 10. In: WHITLEY, Darrell ; WATSON, Jean: *Complexity Theory and the No Free Lunch Theorem*. Springer, 2005, S. 317–339
- [Wolpert 1992] WOLPERT, David H.: Stacked generalization. In: *Neural Networks* 5 (1992), Nr. 2, S. 241 – 259. [http://dx.doi.org/10.1016/S0893-6080\(05\)80023-1](http://dx.doi.org/10.1016/S0893-6080(05)80023-1). – DOI 10.1016/S0893-6080(05)80023-1
- [Ziegler 1997] ZIEGLER, Bart: IBM's Winning 'Deep Blue' Is Still Product of Primates. In: *The Wall Street Journal* (1997). <https://www.wsj.com/articles/SB863381328224169500>. – Besucht: 02. Januar 2020