

BACHELORARBEIT

Boris Thibaut Tondjua

Big-Data-Analysen von Materialstammdaten zur
Ableitung von Regeln für eine automatisierte Daten-
Vorbelegung

Matrikelnummer:	3058341
Fakultät:	Maschinenbau
Studiengang:	Maschinenbau
Abgabefrist:	31.08.2021
Betreuer/Prüfer:	Prof. Dr. rer. Nat. Markus Goldhacker
Zweitprüfer:	Prof. Dipl.-Wirtsch.-Ing. Stefan Galka
Externe Betreuung:	Herr Andre Müller, Krones AG, GDDM

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Implementierung und der Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben. Ein besonderer Dank gilt Prof. Dr. rer. Nat. Goldhacker und Herrn Müller, die meine Arbeit betreut haben.

Darüber hinaus gebührt mein Dank Patrick Kamm für die Betreuung und Begutachtung dieser Bachelorarbeit. Für die hilfreichen Anregungen und die konstruktive Kritik bei der Erstellung dieser Arbeit möchte ich mich herzlich bedanken.

Abschließend möchte ich mich bei meinen Eltern und Freunden bedanken, die mir mein Studium durch ihre Unterstützung ermöglicht haben und stets ein offenes Ohr für mich hatten.

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Abkürzungsverzeichnis	IV
Tabellenverzeichnis	V
1 Einleitung	1
1.1 Ausgangssituation.....	2
1.2 Motivation	2
1.3 Ziel der Arbeit.....	6
1.4 Aufbau der Arbeit.....	7
2 Theoretische Grundlagen	8
2.1 Überwachtes Lernen.....	10
2.1.1 Klassifizierung diskreter Label: Vorhersage	11
2.1.2 Regression: Vorhersage von kontinuierlichem Label	14
2.2 Unüberwachtes Lernen	17
2.2.1 Clustering: Ableitung von Label aus Daten.....	17
2.2.2 Dimensionsreduktion	19
2.3 Rechnungsumgebung und verwendete Materialarten.....	21
3 Explorative Datenanalyse	22
3.1 Arten von Feldern	22
3.1.1 Erster Datensatz	22
3.1.2 Zweiter Datensatz	23
3.1.3 Dritter Datensatz	24
3.1.4 Vierter Datensatz	25
3.2 Identifizierung von fehlenden Werten	25
4 Datenvorverarbeitung	28
4.1 Ziel der Datenvorverarbeitung	28
4.2 Imputation der fehlenden Daten	31
4.3 Kodierung von kategorischen Features	31
4.4 Normalisierung von Numerischen Features.....	32
4.5 Pipeline von Features.....	33
5 Untersuchung der Struktur der Daten	34
5.1 Die Ellbogen-Methode	34
6 Modellierung	36
6.1 Anwendung vom überwachten Lernen	37

6.1.1	Der Complement Naive Bayes Algorithmus (CNB).....	38
6.1.2	Der Random Forest Classifier Algorithmus (RFC).....	41
6.2	Gütemaße für die Auswertung der Algorithmen	44
7	Auswertung der Ergebnisse der vier Datensätze	48
7.1	Auswertung der Ergebnisse des ersten Datensatzes.....	48
7.1.1	Auswertung der Ergebnisse aus dem RFC – Algorithmus ...	48
7.1.2	Auswertung der Ergebnisse aus dem CNB – Algorithmus...	50
7.2	Auswertung der Ergebnisse des zweitens Datensatzes	52
7.2.1	Auswertung der Ergebnisse aus dem RFC – Algorithmus ...	52
7.2.2	Auswertung der Ergebnisse aus dem CNB – Algorithmus...	54
7.3	Auswertung der Ergebnisse des dritten Datensatzes	56
7.3.1	Auswertung der Ergebnisse aus dem RFC – Algorithmus ...	56
7.3.2	Auswertung der Ergebnisse aus dem CNB – Algorithmus...	58
7.4	Auswertung der Ergebnisse des viertens Datensatzes.....	59
7.4.1	Auswertung der Ergebnisse aus dem RFC – Algorithmus ...	60
7.4.2	Auswertung der Ergebnisse aus dem CNB – Algorithmus...	62
8	Fazit und Ausblick.....	64
9	Literaturverzeichnis.....	66
	Anhang.....	68
a)	Anhang	68

Abbildungsverzeichnis

Abbildung 1: Ausschnitt des Regelwerks, das bereits zur Anwendung kommt ...	6
Abbildung 2: Einfacher Datensatz zur Erklärung der Klassifizierung (VanderPlas, 2016, S. 333).	11
Abbildung 3: Darstellung eines einfachen Klassifizierungsmodells (VanderPlas, 2016, S. 334).	12
Abbildung 4: Anwendung eines Klassifizierungsmodells auf neue Daten (VanderPlas, 2016, S. 334).	13
Abbildung 5: Einfacher Datensatz für die Erklärung der Regression (VanderPlas, 2016, S. 336).	14
Abbildung 6: Dreidimensionale Darstellung der Regressionsdaten (VanderPlas, 2016, S. 337).	15
Abbildung 7: Darstellung des Regressionsmodells (VanderPlas, 2016, S. 337).	16
Abbildung 8: Anwendung des Regressionsmodells auf neue Daten (VanderPlas, 2016, S. 337).	16
Abbildung 9: Beispieldatei für Clustering (VanderPlas, 2016, S. 339).	18
Abbildung 10: Daten, die mit einem k-Means-Clustermodell klassifiziert sind (VanderPlas, 2016, S. 340).	19
Abbildung 11: Beispieldaten für die Dimensionsreduktion (VanderPlas, 2016, S. 341).	20
Abbildung 12: Daten mit einem durch Dimensionsreduktion erlernten Label (VanderPlas, 2016, S. 341).	21
Abbildung 13: Kreisdiagramm für die MTART 1010	23
Abbildung 14: Kreisdiagramm für die MTART 1000	23
Abbildung 15: Kreisdiagramm für die MTART 1030	24
Abbildung 16 : Kreisdiagramm für die MTART 1040	25
Abbildung 17 : Korrelationsmatrix der MTART 1010 vor der Datenvorverarbeitung	26
Abbildung 18 : Korrelationsmatrix der MTART 1000 vor der Datenvorverarbeitung	26
Abbildung 19 : Korrelationsmatrix der MTART 1030 vor der Datenvorverarbeitung	27
Abbildung 20 : Korrelationsmatrix der MTART 1040 vor der Datenvorverarbeitung	27
Abbildung 21 : Korrelationsmatrix der MTART 1010 nach der Datenvorverarbeitung	29
Abbildung 22 : Korrelationsmatrix der MTART 1000 nach der Datenvorverarbeitung	29
Abbildung 23 : Korrelationsmatrix der MTART 1030 nach der Datenvorverarbeitung	30
Abbildung 24 : Korrelationsmatrix der MTART 1040 nach der Datenvorverarbeitung	30
Abbildung 25: Ellbogen-Methode für die MTART 1010	34

Abbildung 26 : Ellbogen-Methode für die MTART 1000	35
Abbildung 27 : Ellbogen-Methode für die MTART 1030	35
Abbildung 28 : Ellbogen-Methode für die MTART 1040	36
Abbildung 29: Darstellung eines Entscheidungsbaumes (Hatwell et al., 2020, S. 5755).	42

Abkürzungsverzeichnis

m	Anzahl der Zeilen der Matrix X
n	Anzahl der Spalten der Matrix X
$J(\theta)$	Kostenfunktion
θ	Vektor der Parameter des Modells
ML	Maschinelles Lernen
MTART	Materialart
μ_x	Mittelwert der Variable X ist
σ_x	die Standardabweichung der Variablen X ist
k	Anzahl der Cluster
CNB	Complement Naives Bayes
RFC	Random Forest Classifier
DT	Decision Trees
c	Klasse in der Klassifizierung
α_i	Parameter
α	Parameter
$\hat{\theta}_{ci}$	Schätzungen von CNB
$\hat{w}_{\tilde{c}i}$	Gewichte für jede Schätzung der CNB
f_i	Häufigkeit des Wortes i im Dokument d
d	Dokument
CART	Classification and Regression Trees
I_G	Gesamt-Gini-Unreinheit
p_k	Anteil der Instanzen mit dem Label y_k
TN	True Negative
TP	True Positive

<i>FN</i>	Falsch Negativ
<i>FP</i>	Falsch Positiv
<i>K</i>	Gesamte Anzahl der Klassen

Tabellenverzeichnis

Tabelle 1: Tabelle RFC 1010 Felder-----	49
Tabelle 2: Tabelle CNB 1010 Felder -----	50
Tabelle 3: Zusammenfassung des Targets, die nicht vorhergesagt werden sollen.-----	52
Tabelle 4: Tabelle RFC 1000 Felder-----	53
Tabelle 5: Tabelle CNB 1000 Felder -----	55
Tabelle 6: Zusammenfassung des Targets, die nicht vorhergesagt werden sollen -----	56
Tabelle 7: Tabelle RFC 1030 Felder-----	57
Tabelle 8: Tabelle CNB 1030 Felder -----	59
Tabelle 9: Tabelle RFC 1040 Felder-----	61
Tabelle 10: Tabelle CNB 1040 Felder-----	63
Tabelle 11: Bezeichnungen von den verwendeten Feldern (Features, Target)	68

1 Einleitung

Die vorliegende Bachelorarbeit ist beim Unternehmen Krones AG verfasst worden. Die Krones AG, im Folgenden Krones genannt, ist ein deutsches Unternehmen, welches in der Maschinenbaubranche tätig ist. Krones stellt Anlagen und Maschinen für die Abfüllung und Verpackung von Getränken und flüssigen Nahrungsmitteln her, deckt mit seinen Produkten den kompletten Produktions-, Abfüll- und Verpackungsprozess ab und integriert entsprechende IT-Systeme. Ob aus Glas, PET oder Aluminium, die Maschinen und Anlagen von Krones verarbeiten täglich Millionen an Flaschen, Dosen und Formbehältern. Als Marktführer der Branche zählen zu den Krones Kunden hauptsächlich Brauereien, Wasser-, Softdrink- und Saft-Hersteller, Molkereien, Wein-, Sekt- und Spirituosen-Produzenten sowie Unternehmen der Liquid Food-Branche. Im Konzern gibt es über 10 Mio. Materialstamm-Sätze (= Teile-Stammsätze), die sich je nach Einsatzzweck in verschiedene Materialarten gliedern lassen: Eigenfertigungsteil, Kaufteil, Baugruppe, Teil für interne Betriebsmittel etc. In Summe werden zu diesen Materialstamm-Sätzen als Voraussetzung für die interne und externe Abwicklung (Produktion, Beschaffung, Kalkulation, Logistik, Zoll, Ersatzteilabwicklung ...) mehr als 200 Felder befüllt und gepflegt. Die verwendeten Felder variieren je nach Materialart (ein selbst-produziertes Teil benötigt andere Informationen als ein Katalog-Teil, das zugekauft wird). Die Pflege dieser Felder findet zum einen manuell durch die Anwender statt und zum anderen automatisiert für bestimmte Konstellationen oder Anwendungsfälle. Täglich werden ca. 700 neue Materialstämme angelegt und die benötigten, nicht automatisiert befüllten Felder müssen anschließend manuell von den verschiedensten Abteilungen eingepflegt werden, bevor der Datensatz operativ genutzt werden kann. Dies verursacht einen erheblichen Aufwand.

1.1 Ausgangssituation

Die Arbeit mit großen Datenmengen ist im Laufe der Zeit sehr schwierig geworden.

Um diese Schwierigkeiten zu verringern und die Nutzung dieser Daten zu erleichtern, ist es daher wichtig, Methoden zur Verbesserung und Beschleunigung der Verarbeitung dieser Daten zu entwickeln.

Aus den vielen vernetzten Geräten resultieren enorme Datenmengen, deren schnelle Bearbeitung mit Automatisierung realisierbar ist. So können Unternehmen aus den Rohdaten wertvolle Daten und abgeleitete Erkenntnisse in Echtzeit empfangen, bearbeiten und zur Nutzung bereitstellen. Die im Laufe der Zeit zunehmende Datenmenge in der Industrie hat zum Einsatz von Automatisierungsprozessen geführt, um die darin enthaltenen Informationen besser nutzen zu können. Automatisierung ermöglicht die Verwaltung von enormen Datenvolumen und liefert Erkenntnisse, aus denen ein Unternehmen Wert schöpfen kann. Durch den Entfall manueller Programmierung und der wiederkehrenden zeitaufwändigen Anforderungen von Dateninfrastrukturprojekten reduziert die Automatisierung den Bedarf an menschlicher Interaktion. Dies bringt mehrere entscheidende Vorteile mit sich. Zum Beispiel können Erkenntnisse aus den Daten in kürzerer Zeit und kostengünstiger bei deutlich verbesserter Qualität und Zuverlässigkeit der Ergebnisse geliefert werden. Außerdem bleibt dadurch der Fokus der zuständigen Mitarbeiter auf die eher strategischen Inhalte ihrer Arbeit (Gillhuber, 2019).

1.2 Motivation

Die exponentielle Zunahme von Daten in den letzten Jahren aufgrund menschlicher Aktivitäten in verschiedenen Tätigkeitsbereichen wie sozialen Netzwerken, Online-Verkaufsplattformen und dem Industriesektor hat eine neue Computertechnologie hervorgebracht, die als Big Data bekannt ist.

Big Data ist gegenwärtig bei zahlreichen Unternehmen im Fokus der Aufmerksamkeit, da die Thematik für ihre ökonomische Weiterentwicklung

genutzt werden soll. Der Begriff Big Data bezieht sich auf Datenbestände, die so groß, schnelllebig oder komplex sind, dass sie sich mit herkömmlichen Methoden nicht oder nur schwer verarbeiten lassen.

Das Datenwachstum fördert eine Vielzahl technologischer Innovationen und Kreationen. Das Verständnis der 4 V's von Big Data kann für spezifische Forschungen und reale Probleme der Welt genutzt werden (V.S.Thiyagarajan & K.Venkatachalapathy, 2014, S. 134–135). Im Folgenden werden die 4 V's von Big Data erklärt.

- Volume (das Volumen)

Big Data impliziert enorme Datenmengen. Nun, da die Daten von Maschinen, Netzwerken und menschlicher Interaktion in Systemen wie sozialen Medien erzeugt werden, ist das Volumen der zu analysierenden Daten massiv geworden. Das Volumen von Big Data bezieht sich auf die Größe der Daten, die aus allen Quellen erfasst wurden, beispielsweise aus Text, Audio, Video, sozialen Netzwerken, Forschungsstudien, medizinischen Daten, Bildern, Kriminalitätsberichten, Wettervorhersagen oder Naturkatastrophen. Eingangsdaten für Big Data Systeme können Chats aus sozialen Netzwerken, Webserver, Sensoren, Satellitenbilder, Audioübertragungen, Bankgeschäfte, Banktransaktionen, der Inhalt von Webseiten, Scans von Regierungsdokumenten, GPS-Spuren, Finanzmarktdaten usw. sein. Da solche Datenmengen jedoch ungeordnet und unbekannt sind, können sie nicht mit herkömmlichen Methoden bearbeitet oder abgefragt werden (V.S.Thiyagarajan & K.Venkatachalapathy, 2014, S. 133).

- Velocity (Die Geschwindigkeit)

Die Geschwindigkeit von Big Data, verbunden mit ihrer Vielfalt, wird zu Echtzeit-Beobachtungen führen, die bessere Entscheidungsfindung oder schnelles Handeln ermöglichen. Da sich der Markt weiterentwickelt, ist es wahrscheinlich, dass die meisten dieser Beobachtungen das Ergebnis von Anwendungen sein werden, die die Fähigkeit zur Reaktion auf Veränderungen ermöglichen werden. Analysen werden helfen, diese Anwendungen einfacher und effizienter zu erstellen. Diese hohe Geschwindigkeit ist direkt verantwortlich für das hohe Volumen. Angesichts dieser Geschwindigkeit, mit der die Daten kommen,

müssen die Unternehmen Technologie und Datenbanken vorbereiten, um die Daten bei Bedarf zu verarbeiten (V.S.Thiyagarajan & K.Venkatachalapathy, 2014, S. 134).

- Variety (die Vielfalt)

Daten treten in vielen verschiedenen Formen auf. Deswegen ist es eine große Herausforderung, ein System zu etablieren oder aufzubauen, damit eine solche Datenmischung direkt in das System integriert werden kann. Selten sind Daten vor der Verarbeitung in einer perfekt geordneten und verarbeitungsbereiten Form vorhanden. Ein gemeinsames Thema bei Big-Data-Systemen ist, dass die Quelldaten vielfältig sind und sich nicht in saubere relationale Strukturen befinden. Es wird festgestellt, dass die Vielfalt der Daten sich direkt auf die Integrität der Daten auswirkt. Mit anderen Worten, je mehr Vielfalt in den Daten ist, desto mehr Fehler werden sie enthalten (V.S.Thiyagarajan & K.Venkatachalapathy, 2014, S. 134).

- Value (der Wert)

Im Gegensatz zu anderen V's von Big Data, die bis jetzt besprochen wurden, ist dieses V das gewünschte Ergebnis der Verarbeitung von Big Data. Man wird immer daran interessiert sein, den größtmöglichen Nutzen aus einem großen Datensatz zu ziehen, der zur Verfügung steht. Es muss hier nach dem wahren Wert der Daten gesucht werden, mit denen es bearbeitet werden soll. Der Wert der Daten hängt auch stark von den Verwaltungsmechanismen ab. Das heißt, wie Richtlinien und Strukturen entwickelt werden, die letztendlich ein Gleichgewicht zwischen Nutzen und Risiko der Daten herstellen (V.S.Thiyagarajan & K.Venkatachalapathy, 2014, S. 134).

Daten können strukturiert und unstrukturiert sein. Ein häufiges Problem ist, dass die riesigen Datenmengen nicht für Analysen aufbereitet sind. Ebenfalls sind, beispielsweise durch Kundenaccounts und Registrierungen, doppelte Datensätze möglich. Einige Daten sind einem Eigentümer zugeordnet, während andere durch und mit Unternehmen generierte Datensätze sind oder durch den Anschluss von Geräten wie Sensoren oder Prozessoren entstehen. Ein Unternehmen verarbeitet Daten, deren Wert von ihrer Nutzung, Herkunft und Auswertung abhängig ist. Die Auswertung dieser Daten und die konstante

Optimierung der Ergebnisse haben als Folge die Optimierung der betrieblichen Abläufe betroffener Unternehmen.

Im Kontext von Daten kommt es zum Data Mining, auch Auswertung von großen Datenmengen genannt. Hier ist die Erkennung von Mustern und Zusammenhängen innerhalb der Datensätze das Ziel. Dafür benötigt Big Data verschiedene Analysen, unter anderem das Clustering, die Entscheidungsbäume. In diesem Zusammenhang sind beispielsweise für Produkteinführungen oder zur Marktforschung Analysen sinnvoll und bei der Bewertung von internen und externen Faktoren hilfreich. Im Mittelpunkt steht ein mögliches Szenario, auf dem weitere Maßnahmen, Konzepte, Lösungen und Planungen aufbauen können (MDIS, 2020).

Auch Krones nutzt Big Data und entwickelt ständig Strategien, um seine Daten optimal zu nutzen und ein Maximum an Informationen für das Funktionieren der Systeme herauszuholen, denn das Datenmanagement stellt heute eine große Herausforderung in allen Bereichen der Geschäftswelt dar.

1.3 Ziel der Arbeit

Ziel der Arbeit ist es, mittels Big Data-Analysen Felder (Spalten in einer SAP-Tabelle) zu ermitteln, welche eine gegenseitige Abhängigkeit und eine Vorbelegung von Datensätzen ermöglichen. In der Abbildung 1 ist ein Ausschnitt aus solch einer SAP-Tabelle dargestellt.

MANDT	WERK	ZZTEXT	MATERIAL	ZZSTUELLART	LABOR	DISPO	MELDE	DISPO	DISPOL	FESTE	BESCH	SON	PROD	FREM	EIGENF	PLANU	HORIZO	PERIO	STRATE	VERFU	EINZEL	PRODU
						MERKM	BESTAN	NENT	OSGR	LOS	HAFU	DER	UKTI	DBES	ERTIGU	EFE	NTENS	DE	GIEGRU	EGBAR	SAMM	KTION
						AL	D		OESSE	ROES	UNGS	BES	ONSL	CHLA	NGSZ	ER	CHL	NNZEI	PPE	KEITS	EL	ST
100	1001	00006	1000		25C	PD	0,000	FB1	EX	0,000	E		1599	1103	7	0	FP1	M	20	02	1	17X
100	1001	00006	1000		LAE	PD	0,000	PB1	EX	0,000	E		1599	1103	7	21	FP1	M	20	02	1	17X
100	1001	00006	1030		25C	PD	0,000	FB1	EX	0,000	E		1599	1103	7	0	FP2	M	20	02	1	17X
100	1001	00006	1030		LAE	PD	0,000	PB1	EX	0,000	E		1599	1103	7	21	FP2	M	20	02	1	17X
100	1001	00190	1030		25A	PD	0,000	FB1	EX	0,000	F		0100	1103	0	21	FP2	M	20	02	1	1
100	1001	00190	1030		LAE	PD	0,000	FB1	EX	0,000	E		1339	1301	12	21	FP2	M	20	02	1	3
100	1001	00485	1030	31	*	PD	0,000	EB1	ZZ	0,000	E		0100		10	0	FP2	M	41	02	2	10E
100	1001	00485	1030	32	*	PD	0,000	EB1	ZZ	0,000	E		0100		10	0	FP2	M	41	02	2	10E
100	1001	00618	1000		25C	PD	0,000	FB1	ZZ	0,000	E		1599	1103	7	0	FP1	M	41	02	2	17X
100	1001	00618	1000		LAE	PD	0,000	PB1	EX	0,000	E		1599	1103	12	21	FP1	M	41	02	2	17X
100	1001	00899	1030	31	*	PD	0,000	EB1	EX	0,000	E		0100		10	0	FP2	M	20	02	1	10E
100	1001	00899	1030	32	*	PD	0,000	EB1	EX	0,000	E		0100		10	0	FP2	M	20	02	1	10E
100	1001	00967	1000		25C	PD	0,000	FB1	ZZ	0,000	F		0100	1102	0	21	FP2	M	41	02	2	1
100	1001	00967	1000		C*	PD	0,000	FB1	ZZ	0,000	F		0100	1502	0	21	FP2	M	41	02	2	1
100	1001	00967	1000		LAE	PD	0,000	PB1	ZZ	0,000	F		0100	1102	0	21	FP2	M	41	02	2	1
100	1001	01083	1030	31	*	PD	0,000	KB1	EX	0,000	E		1599		15	0	FP2	M	20	02	1	14X
100	1001	01083	1030	32	*	PD	0,000	KB1	EX	0,000	E		1599		15	0	FP2	M	20	02	1	14X
100	1001	01354	1030	31	AA*	PD	0,000	EB1	ZZ	0,000	E		0100		10	0	FP1	M	41	02	2	10X
100	1001	01354	1030	32	AA*	PD	0,000	EB1	ZZ	0,000	E		0100		10	0	FP1	M	41	02	2	10X

Abbildung 1: Ausschnitt des Regelwerks, das bereits zur Anwendung kommt

Die grünen Spalten stellen dabei die Features dar. Mithilfe einer Vorbelegung dieser Features können alle weiteren grauen Felder (Target) mit Werten vorhergesagt und gefüllt werden. Die Features und Targets sind in der vorliegenden Bachelorarbeit durch Maschinelles Lernen Algorithmen und Big Data Analysen ermittelt worden.

1.4 Aufbau der Arbeit

Im ersten Teil (Abschnitt 2) der vorliegenden Arbeit geht es sowohl um die theoretischen Grundlagen als auch um die Definition wichtiger Begriffe, die für das Verständnis des Themas relevant sind. In den Unterabschnitten 2.1 und 2.2 liegt der Fokus zunächst auf dem Erklären von überwachtem und unüberwachtem Lernen.

Die Präsentation der in dieser Arbeit verwendeten Rechnungsumgebung, Programmiersprache und Materialarten für den Aufbau der Datensätze wird im Unterabschnitt 2.3 dargestellt.

Der Abschnitt 3 dieser Arbeit befasst sich mit der explorativen Datenanalyse. Es geht hier darum, die Daten so gut wie möglich zu verstehen und einige erste Informationen aus den Daten zu gewinnen.

Der Abschnitt 4 der vorliegenden Arbeit betrifft die Datenvorverarbeitung. Es ist der Schritt, der darin besteht, Daten vorzubereiten, bevor sie dem Algorithmus zur Verfügung gestellt werden.

Im Abschnitt 5 wird die Struktur der Daten mit Hilfe der Ellbogen-Methode untersucht.

Nach der Erarbeitung der oben erwähnten Kapitel kommt die Modellierung im Abschnitt 6. Es wird hier ein Modell von maschinellem Lernen entwickelt und optimiert und zum Schluss werden im Abschnitt 7 die Ergebnisse ausgewertet.

2 Theoretische Grundlagen

Maschinelles Lernen ist ein Unterbereich der künstlichen Intelligenz. Der Schwerpunkt liegt dabei auf dem Trainieren von Algorithmen, um aus Daten und Erfahrungen zu lernen und diese stets zu verbessern – anstatt explizit dafür programmiert zu werden. Beim maschinellen Lernen werden Algorithmen darauf trainiert, Muster und Korrelationen in großen Datensätzen zu finden und auf Basis dieser Analyse die besten Entscheidungen und Vorhersagen zu treffen. Anwendungen für maschinelles Lernen verbessern sich mit der Nutzung und werden theoretisch umso genauer, je mehr Daten sie zur Verfügung haben. Davon betroffen sind viele Bereiche des täglichen Lebens, unter anderem der Haushalt, das Einkaufserlebnis, die Unterhaltungsbranche sowie das Gesundheitswesen.

Maschinelles Lernen setzt sich aus verschiedenen Arten von maschinellen Lernmodellen zusammen, die verschiedene algorithmische Techniken verwenden. Abhängig von der Art der Daten und dem gewünschten Ergebnis kann einer von vier Bereichen des maschinellen Lernens (überwacht, unüberwacht, teilüberwacht oder verstärkend) genutzt werden und sich innerhalb eines Modells mehrere algorithmische Techniken ergeben. Algorithmen des maschinellen Lernens sind grundsätzlich dafür ausgelegt, Dinge zu klassifizieren, Muster zu finden, Ergebnisse vorherzusagen und Entscheidungen zu treffen. Die Algorithmen können einzeln oder kombiniert eingesetzt werden, um bei komplexen und unvorhersehbaren Daten die bestmögliche Genauigkeit zu erzielen (SAP, o.J.).

Die Analyse hochdimensionaler Daten ist eine Herausforderung auf dem Gebiet des maschinellen Lernens und des Data Mining. Die Feature-Selektion (Auswahl von Features) bietet eine effektive Möglichkeit, dieses Problem zu lösen, indem es irrelevante und redundante Daten entfernt, was die Berechnungszeit verkürzt, das Lernen des Modells verbessert und ein besseres Verständnis der Daten für das Modell ermöglicht.

Um maschinelles Lernen besser zu verstehen, sollen im Folgenden einige Begriffe erläutert werden:

- Der Datensatz:

Tabelle (X, y) , die 2 Arten von Variablen enthält:

- Target y
- Features X

m ist die Anzahl der Beispiele, die die Tabelle enthält (die Anzahl der Zeilen) und n die Anzahl der Features (die Anzahl der Spalten von X).

- X ist eine Matrix mit m Zeilen und n Spalten. $X \in \mathbb{R}^{m \times n}$
- y ist ein Vektor (ein Spaltenvektor) mit m Zeilen. $y \in \mathbb{R}^m$

Um das Feature j des Beispiels i zu bezeichnen, wird $x^{(i)}$ geschrieben (Saint-Cirgue, 2019, S. 94).

- Das Modell:

Bezeichnet eine mathematische Funktion, die X mit y verknüpft, sodass $f(X) = y$ entsteht. Ein gutes Modell sollte eine gelungene Verallgemeinerung sein, d. h. es sollte Folgendes bieten: kleine Fehler zwischen $f(x)$ und y , ohne dass es zu einer Überanpassung kommt. Im Kontext des maschinellen Lernens wird von Überanpassung gesprochen, wenn der Algorithmus im Prinzip den Datensatz auswendig lernt, aber nicht das zugrundeliegende Muster oder System erkennen kann. Damit sind Prognosen, die der Algorithmus aus noch unbekannten Daten liefern soll, nicht gut (Saint-Cirgue, 2019, S. 95).

- Die Kostenfunktion:

Die Kostenfunktion $J(\theta)$ misst die Menge der Fehler zwischen dem Modell und dem Datensatz. θ ist der Vektor, der die Parameter unseres Modells enthält (Saint-Cirgue, 2019, S. 95).

- Die Varianz:

Dies ist der Fehler, der auf ein Modell zurückzuführen ist, das zu sehr auf Details reagiert und nicht verallgemeinern kann, was zu einem Over-fitting (Überanpassung) führt (Saint-Cirgue, 2019, S. 96).

- Das Bias (Voreingenommenheit):

Dies ist der Fehler, der auf ein fehlerhaftes Modell zurückzuführen ist, dem es an Präzision mangelt und das zu einem Under-fitting (Unteranpassung) führt (Saint-Cirgue, 2019, S. 96).

Im maschinellen Lernen wird oftmals mit unausgewogenen Datensätzen gearbeitet. Unausgewogene Datensätze sind Datensätze, bei denen die Anzahl der Beispiele einer Klasse höher ist als die Anzahl der Beispiele, die zu anderen Klassen gehören. Das bedeutet, dass die Verteilung der Beispiele nicht gleichmäßig ist. Die Arbeit mit dieser Art von Datensätzen kann schwierig sein, da ein Modell diese Daten leicht zugunsten der Klasse mit der größeren Anzahl von Beispielen überanpassen kann.

In den folgenden Abschnitten 2.1 und 2.2 werden die zwei häufigsten Methoden des maschinellen Lernens erklärt.

2.1 Überwachtes Lernen

Im Bereich des überwachten Lernens existiert eine sogenannte Grundwahrheit. Es gibt Trainingsdaten, bei denen die Features sowie die Label bekannt sind. Aus diesen Daten werden Modelle erstellt, die das Ergebnis liefern. Nach der Erstellung der Modelle können unbekannte Daten zur Verfügung gestellt werden, mit denen das System das Ergebnis berechnet (Paluv, 2018).

Das wird folgendermaßen mathematisch formuliert:

- X ist eine Feature-Matrix mit m Zeilen und n Spalten. $X \in \mathbb{R}^{m \times n}$ mit der Dimension $m \times n$.
- y ist das zugehörige Label, ein Vektor (Spaltenvektor) mit m Zeilen. $y \in \mathbb{R}^m$ mit der Dimension m .

- Der Trainingsdatensatz ist in diesem Fall ein Datensatz mit Beispielen (oder auch Labels genannt), die für das Lernen der Muster und Zusammenhänge in den Daten verwendet wird. Für jedes Sample oder Instanz (Zeile in der Feature Matrix) gibt es einen entsprechenden Eintrag in \mathcal{Y} .

Das überwachte Lernen wird weiter in Klassifizierungs- und Regressionsaufgaben unterteilt: Bei der Klassifizierung sind die Labels diskrete Kategorien, während bei der Regression die Labels kontinuierlich sind. Für die Klassifizierung werden diskrete Labels (zwischen diesen Labels kann kein Abstandsbegriff definiert werden) und für die Regression werden kontinuierliche Labels (die Abstandsbegriffe zwischen diesen Labels haben eine Bedeutung) vorhergesagt.

2.1.1 Klassifizierung diskreter Label: Vorhersage

Zunächst wird eine einfache Klassifizierungsaufgabe betrachtet, die eine Reihe von markierten Punkten enthalten und zur Klassifizierung einiger nicht markierter Punkte verwendet werden. Stellen Sie sich vor, Sie haben die Daten der Abbildung 2.

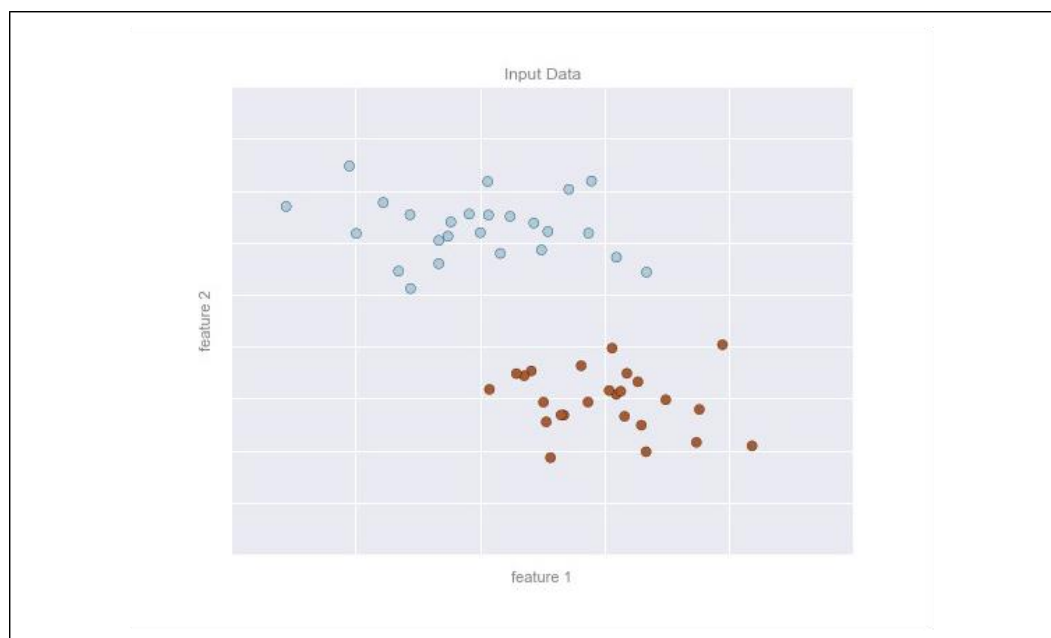


Abbildung 2: Einfacher Datensatz zur Erklärung der Klassifizierung (VanderPlas, 2016, S. 333).

Wir haben in der Abbildung 2 zweidimensionale Daten, d. h. wir haben zwei Features für jeden Punkt, die durch die (x, y) -Positionen der Punkte auf der Ebene dargestellt werden. Außerdem haben wir für jeden Punkt eine von zwei Klassen von Labels, hier dargestellt durch die Farben der Punkte. Aus diesen Features und Labels wird ein Modell erstellt, mit dem entschieden wird, ob ein neuer Punkt als "blau" oder "rot" bezeichnet werden soll. Es wird davon ausgegangen, dass die beiden Gruppen durch das Ziehen einer geraden Linie durch die Ebene zwischen ihnen getrennt werden können, sodass die Punkte auf jeder Seite der Linie in dieselbe Gruppe fallen. In diesem Fall ist das Modell eine quantitative Version der Aussage "eine gerade Linie trennt die Klassen", während Modellparameter bestimmte Zahlen sind, die die Lage und Ausrichtung dieser Linie für die Daten beschreiben. Die optimalen Werte für diese Modellparameter werden aus den Daten erlernt (dies ist das "Lernen" beim maschinellen Lernen), was oft als Training des Modells bezeichnet wird (VanderPlas, 2016, S. 334).

Wie eine visuelle Darstellung als trainiertes Modell für diese Daten aussehen kann, wird in der Abbildung 3 deutlich.

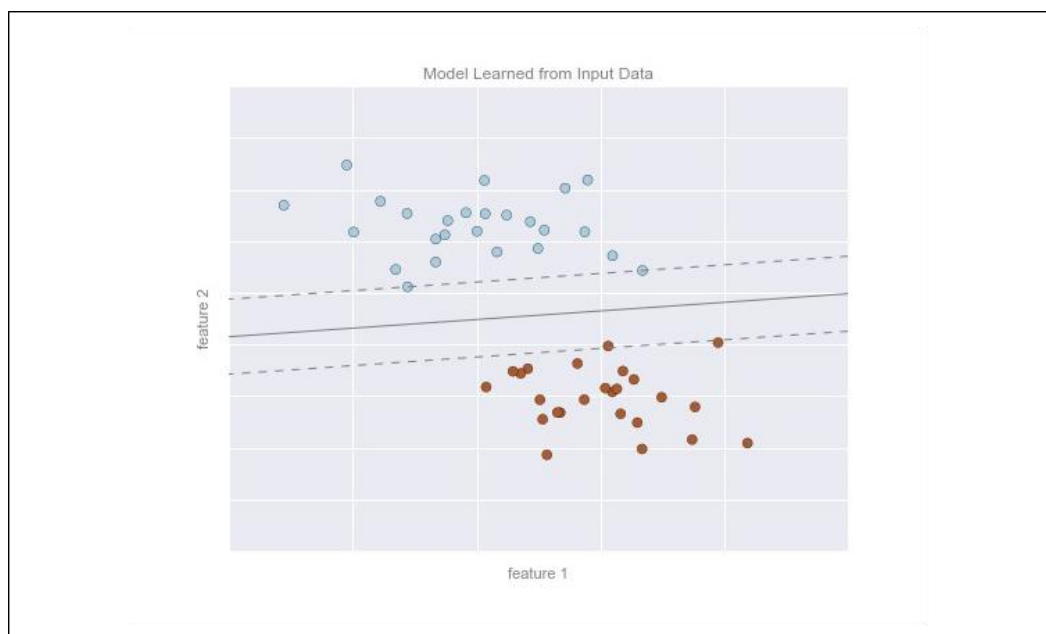


Abbildung 3: Darstellung eines einfachen Klassifizierungsmodells (VanderPlas, 2016, S. 334).

Nach dem Training des Modells, kann es auf neue Daten, die keine Label haben, verallgemeinert werden. Das heißt, dass ein neuer Datensatz hinzugezogen

werden kann, um diese Modelllinie durch ihn zu ziehen und den neuen Punkt auf der Grundlage dieses Modells Label zuzuweisen. Diese Phase wird gewöhnlich als Vorhersage bezeichnet und ist in der Abbildung 4 dargestellt.

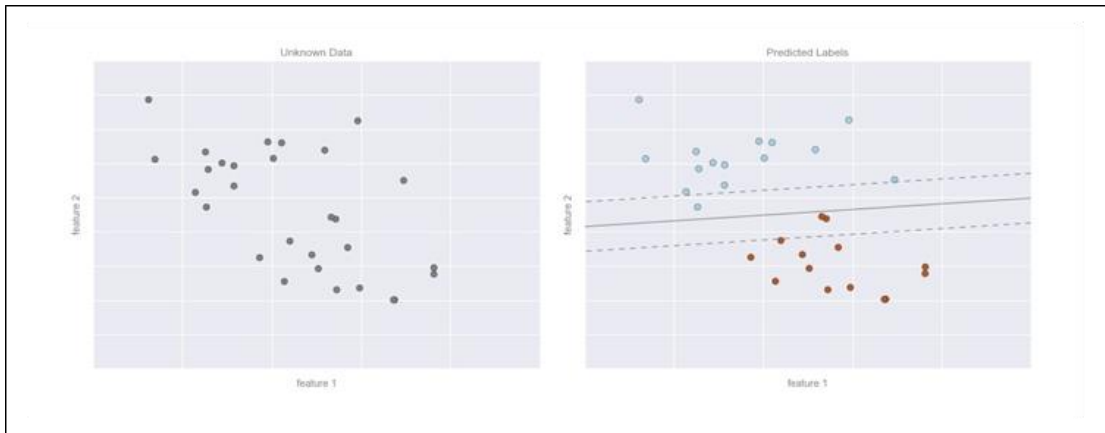


Abbildung 4: Anwendung eines Klassifizierungsmodells auf neue Daten (VanderPlas, 2016, S. 334).

Dies ist die Grundidee einer Klassifizierungsaufgabe beim maschinellen Lernen, wobei "Klassifizierung" bedeutet, dass die Daten diskrete Label haben (VanderPlas, 2016, S. 334–335).

Zwischen den Klassifizierungsaufgaben gibt es die binären Klassifizierungsaufgaben (eine Klassifizierungsaufgabe mit nur zwei Klassen) und die Multiklassenklassifizierung (eine Klassifizierungsaufgabe mit mehr als zwei Klassen).

2.1.2 Regression: Vorhersage von kontinuierlichem Label

Als Nächstes wird eine einfache Regressionsaufgabe erklärt, bei der die Label kontinuierliche Größen sind.

Betrachtet werden die in der Abbildung 5 gezeigten Daten, die aus einer Reihe von Punkten bestehen, die jeweils mit einem kontinuierlichen Label versehen sind.



Abbildung 5: Einfacher Datensatz für die Erklärung der Regression (VanderPlas, 2016, S. 336).

Wie beim Klassifizierungsbeispiel haben wir zweidimensionale Daten, d. h. es gibt zwei Features, die jeden Datenpunkt beschreiben. Die Farbe jedes Punktes stellt die kontinuierlichen Label für diesen Punkt dar. Es wird zur Erklärung eine einfache lineare Regression verwendet, um die Punkte vorherzusagen. Dieses einfache lineare Regressionsmodell geht davon aus, dass eine Ebene an die Daten angepasst werden kann, wenn das Label als dritte räumliche Dimension betrachtet wird. Dies ist eine übergeordnete Verallgemeinerung des Problems der Anpassung einer Linie an Daten mit zwei Koordinaten (VanderPlas, 2016, S. 336). Dieser Aufbau kann in der Abbildung 6 veranschaulicht werden.

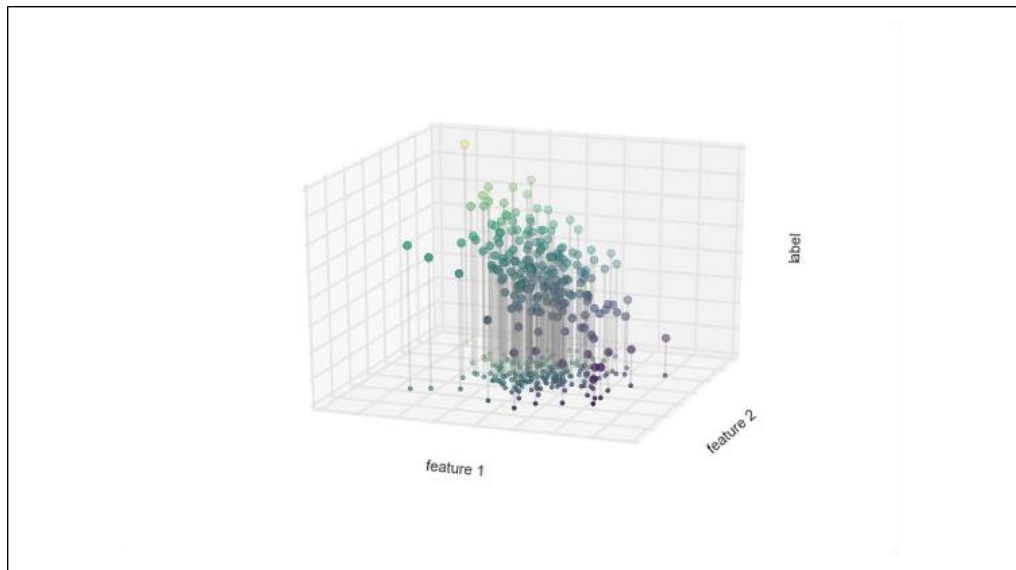


Abbildung 6: Dreidimensionale Darstellung der Regressionsdaten (VanderPlas, 2016, S. 337).

Es wird betrachtet, dass die Ebene von Feature 1 bis Features 2 dieselbe ist wie in der zweidimensionalen Darstellung der Abbildung 5; in diesem Fall wurden jedoch die Label sowohl durch Farbe als auch durch die Position der dreidimensionalen Achse dargestellt. Aus dieser Sicht scheint es vernünftig, dass die Anpassung einer Ebene durch diese dreidimensionalen Daten es ermöglichen würde, die erwarteten Label für jeden Satz von Features vorherzusagen. Wenn wir zur zweidimensionalen Projektion zurückkehren und eine solche Ebene einpassen, erhalten wir das Ergebnis, das in der Abbildung 7 zu sehen ist (VanderPlas, 2016, S. 337).

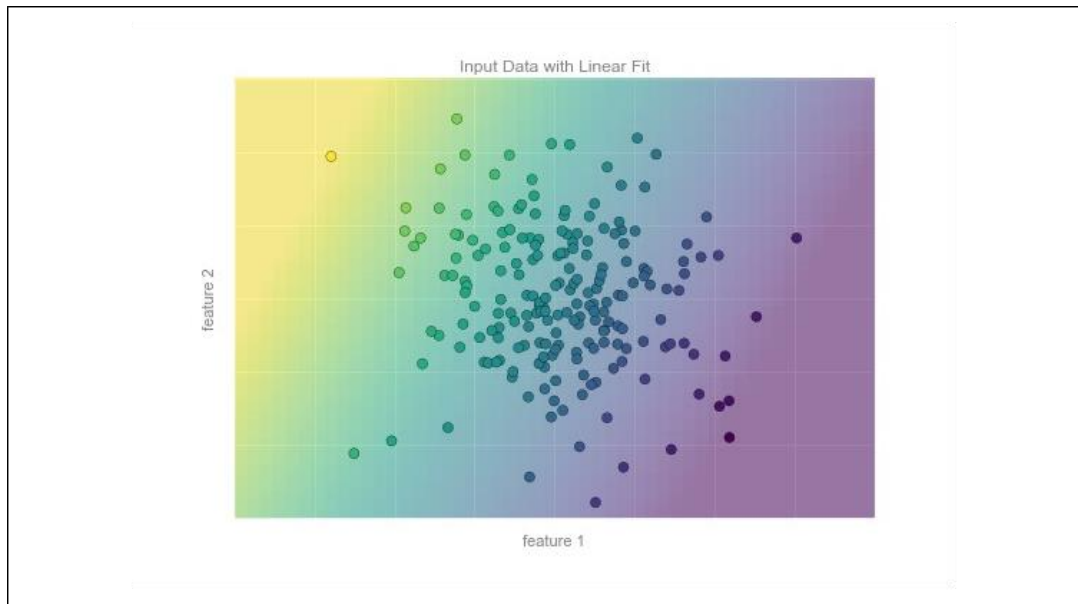


Abbildung 7: Darstellung des Regressionsmodells (VanderPlas, 2016, S. 337).

Diese Anpassungsebene gibt uns das, was gebraucht wird, um die Label für neue Punkte vorherzusagen. Visuell finden wir die Ergebnisse in der Abbildung 8.



Abbildung 8: Anwendung des Regressionsmodells auf neue Daten (VanderPlas, 2016, S. 337).

2.2 Unüberwachtes Lernen

Unüberwachtes Lernen ist der zweite der vier Bereiche des maschinellen Lernens. In diesem ML-Bereich haben die Daten keine Label. Die Maschine untersucht die Eingabedaten und beginnt mit der Erkennung von Mustern und Korrelationen unter Verwendung aller relevanten Informationen in den Daten. Unüberwachtes Lernen ist in vielerlei Hinsicht davon geprägt, wie Menschen die Welt beobachten. Wir nutzen Intuition und Erfahrung, um Dinge zusammenzufassen. Sobald wir mehr und mehr Erfahrungen sammeln, wird unsere Fähigkeit, Erlebnisse zu kategorisieren und zu identifizieren, immer genauer. Für Maschinen wird „Erfahrung“ durch die Menge an Daten definiert, die eingegeben und zur Verfügung gestellt wird. Bekannte Beispiele für Anwendungen des unüberwachten Lernens sind unter anderem Gesichtserkennung, Gensequenzanalyse, Marktforschung und Cybersicherheit (SAP, o.J.).

Dieser Bereich umfasst Algorithmen wie Clustering und Dimensionsreduktion. Im Abschnitt 2.2.1 und 2.2.2 werden jeweils das Clustering und die Dimensionsreduktion erläutert.

2.2.1 Clustering: Ableitung von Label aus Daten

Ein häufiger Fall von unüberwachten Lernen ist das Clustering, bei dem Daten automatisch einer bestimmten Anzahl von diskreten Gruppen zugeordnet werden (VanderPlas, 2016, S. 339). Es können zum Beispiel zweidimensionale Daten betrachtet werden, wie in der Abbildung 9.

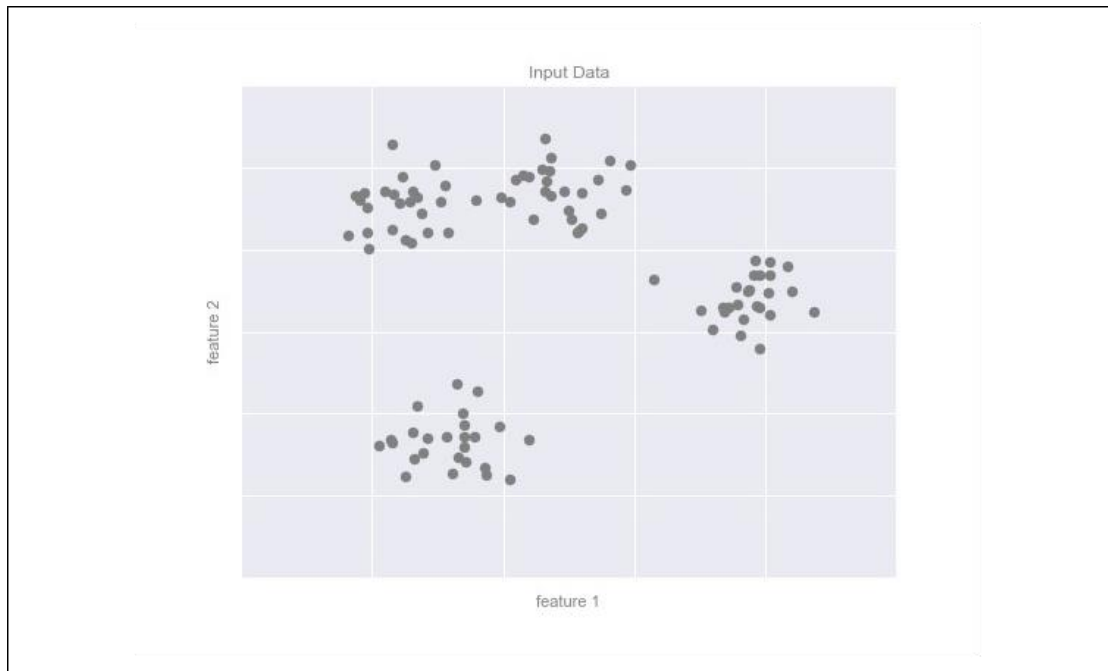


Abbildung 9: Beispieldatei für Clustering (VanderPlas, 2016, S. 339).

Für das Auge ist es klar, dass jeder dieser Punkte zu einer bestimmten Gruppe gehört. Angesichts dieser Eingabe nutzt ein Clustering-Modell die innere Struktur der Daten, um zu bestimmen, welche Punkte miteinander verbunden sind (VanderPlas, 2016, S. 339). Der k -Means-Clustering-Algorithmus ist eine der Clustering-Methoden, die Daten aus einer Menge in Cluster k (k ist die Anzahl von Clustern) partitioniert. Es ist ein distanzbasierter Algorithmus, der die Daten in eine Reihe von Clustern unterteilt.

k -Means-Clustering-Algorithmus passt sich einem Modell an, das aus k Clusterzentren besteht; es wird angenommen, dass die optimalen Zentren diejenigen sind, die den Abstand jedes Punktes von seinem zugewiesenen Zentrum minimieren. Nach der Anwendung des Clustering erhält man die Ergebnisse der Abbildung 10 (VanderPlas, 2016, S. 339).

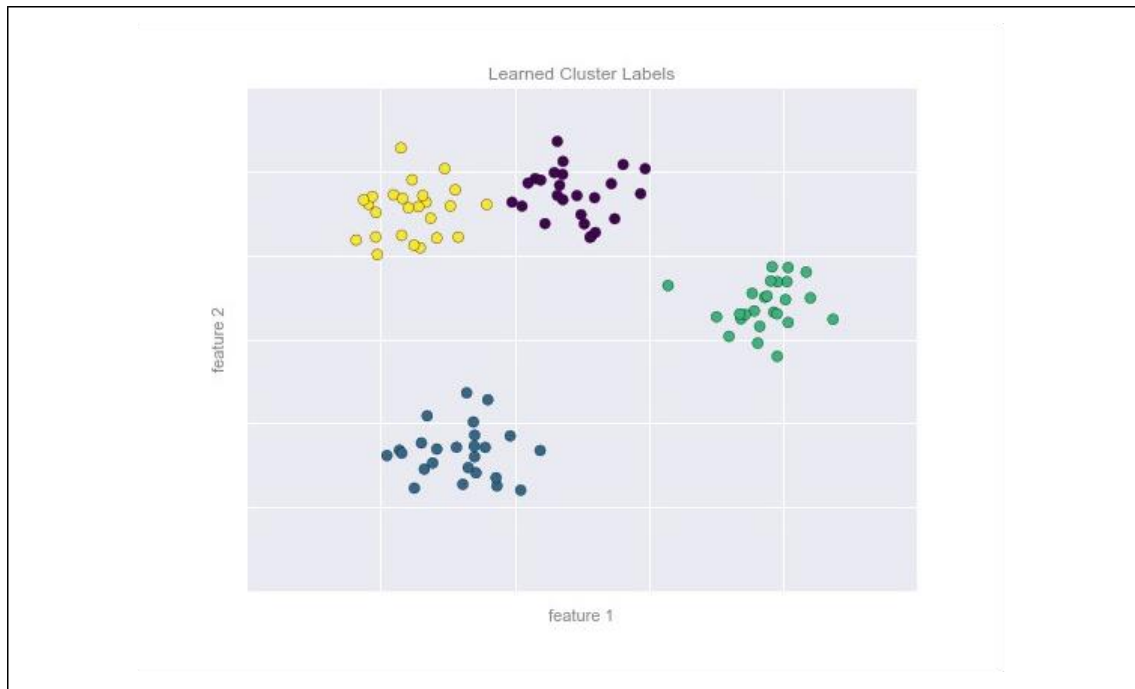


Abbildung 10: Daten, die mit einem k-Means-Clustermodell klassifiziert sind (VanderPlas, 2016, S. 340).

2.2.2 Dimensionsreduktion

Die Dimensionsreduktion ist ein weiteres Beispiel für einen unüberwachten Algorithmus, bei dem Label oder andere Informationen aus der Struktur des Datensatzes selbst abgeleitet werden. Im Allgemeinen wird versucht, eine niedrigdimensionale Darstellung der Daten zu finden, die in gewisser Weise die relevanten Eigenschaften des vollständigen Datensatzes bewahrt (VanderPlas, 2016, S. 340).

In der Abbildung 11 kann auf den ersten Blick visuell erkannt werden, dass diese Daten eine gewisse Struktur aufweisen: Sie stammen aus einer eindimensionalen Linie, die spiralförmig angeordnet ist.

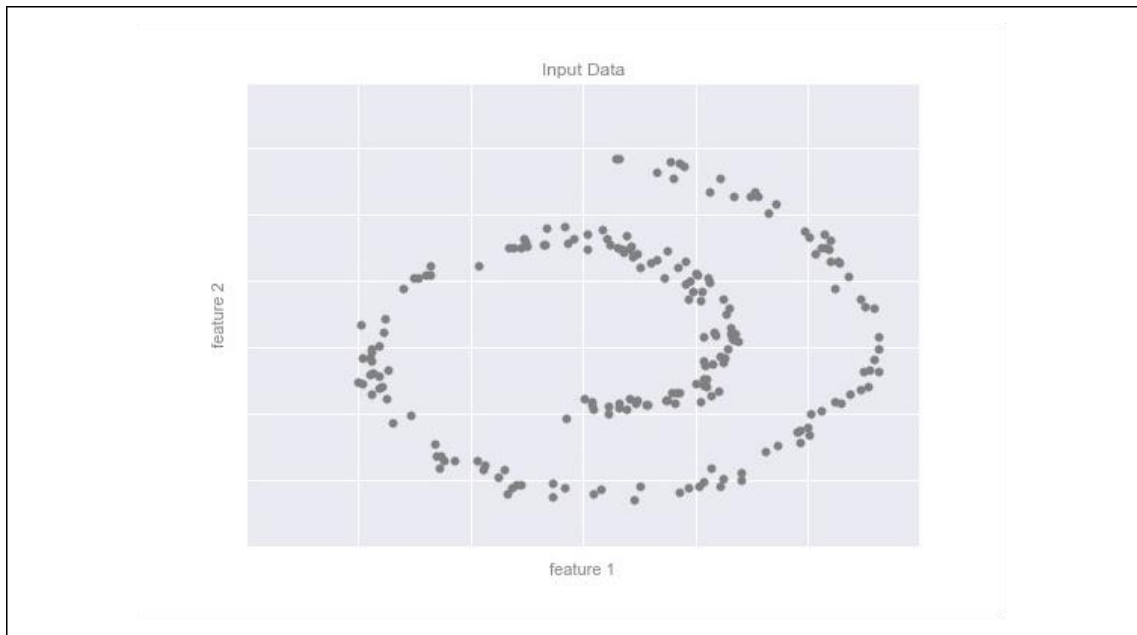


Abbildung 11: Beispieldaten für die Dimensionsreduktion (VanderPlas, 2016, S. 341).

In der Abbildung 12 kann festgestellt werden, dass sich die Farben gleichmäßig entlang der Spirale verändern, was darauf hindeutet, dass der Algorithmus die Struktur der Daten, tatsächlich erkannt hat. Die Farbe stellt eine latente Variable dar. Es ist eine Variable, die nicht direkt beobachtet wird, sondern (durch ein mathematisches Modell) aus anderen beobachteten (direkt gemessenen) Variablen abgeleitet wird. In der Abbildung 12 handelt es sich um eine neue Koordinatenachse (VanderPlas, 2016, S. 341).

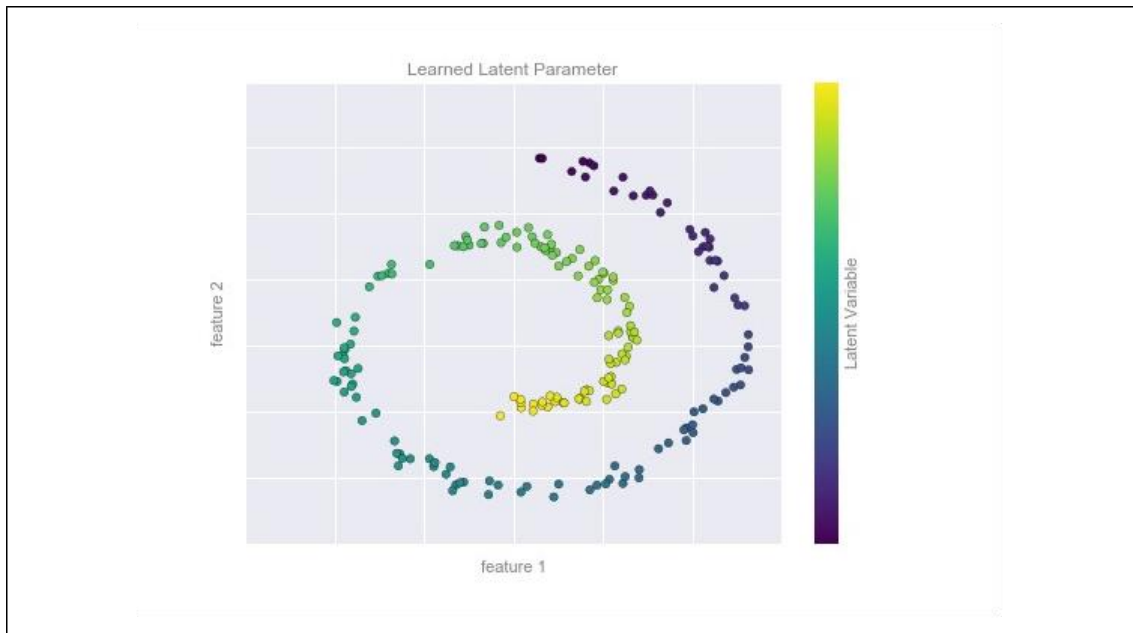


Abbildung 12: Daten mit einem durch Dimensionsreduktion erlernten Label (VanderPlas, 2016, S. 341) .

2.3 Rechnungsumgebung und verwendete Materialarten.

Für den Aufbau der Datensätze wurden die Werte von sieben verschiedenen SAP-Tabellen des Konzerns verwendet und je nach Materialarten miteinander verbunden. Die Daten aus diesen Tabellen wurden mit SparkSQL aus dem Data Lake (Datenbank für die Tabelle des Konzerns) ins Apache Spark Version v3.0.1 extrahiert, mit Python 3.0 bearbeitet und daraus vier verschiedene Datensätze mit jeweils einem Datensatz pro Materialart erstellt.

Es wurden folgende Materialarten betrachtet:

- 1000 Krones Zeichnungsteil
- 1010 Kaufteil
- 1030 Feste Baugruppe
- 1040 konfigurierbares Material

3 Explorative Datenanalyse

Die explorative Datenanalyse verfolgt das Ziel, die Daten so gut wie möglich mit Hilfen von statistischen Methoden wie Verteilungen, Korrelationen zu verstehen und daraus die ersten wichtigen Informationen für die weitere Analyse der Datensätze zu gewinnen.

3.1 Arten von Feldern

In den Datensätzen gibt es viele verschiedene Arten von Variablen bzw. Feldern. Dazu zählen zum Beispiel:

- Numerische Felder sind Felder des Typen float oder integer. Die Abstandsbegriffe in diesen Feldern haben eine Bedeutung.
- Kategorische oder Objektfelder sind Felder des Typen String bzw. Objekt. In diesen Feldern kann kein Abstandsbegriff definiert werden.
- Datetime64(ns) Felder nehmen Bezug auf einen bestimmten Moment in der Zeit, zum Beispiel Datum, Uhrzeit in Stunden, Minuten und Sekunden.

3.1.1 Erster Datensatz

Dieser Datensatz setzt sich aus Feldern zusammen, die nur aus der MTART 1010 des Konzerns stammen. Es wird für die explorative Datenanalyse von einem Datensatz mit 20.000 Zeilen und 147 Feldern ausgegangen. Die Abbildung 13 stellt das Kreisdiagramm der Typen von Feldern und ihre Anteile für diesen Datensatz dar.

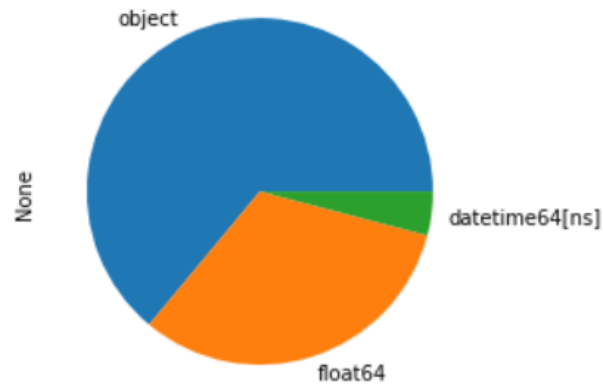


Abbildung 13: Kreisdiagramm für die MTART 1010

3.1.2 Zweiter Datensatz

Dieser Datensatz setzt sich aus Feldern zusammen, die nur aus der MTART 1000 des Konzerns stammen. Für die explorative Datenanalyse wird von einem Datensatz mit 20.000 Zeilen und 195 Feldern ausgegangen. Die Abbildung 14 stellt das Kreisdiagramm der Typen von Feldern und ihre Anteile für diesen Datensatz dar.

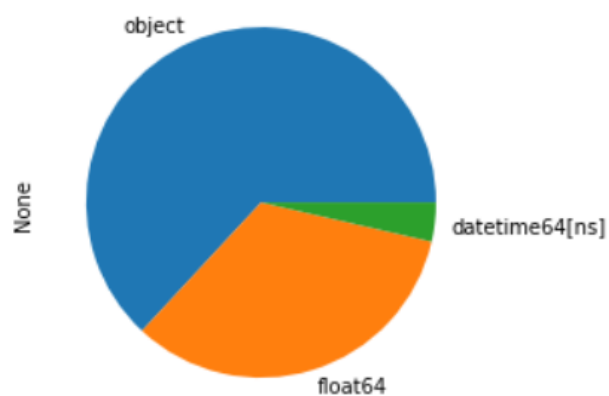


Abbildung 14: Kreisdiagramm für die MTART 1000

3.1.3 Dritter Datensatz

Dieser Datensatz setzt sich aus Feldern zusammen, die nur aus der MTART 1030 des Konzerns stammen. Es wird für die explorative Datenanalyse von einem Datensatz mit 20.000 Zeilen und 165 Feldern ausgegangen. Die Abbildung 15 stellt das Kreisdiagramm der Typen von Feldern und ihre Anteile für diesen Datensatz dar.

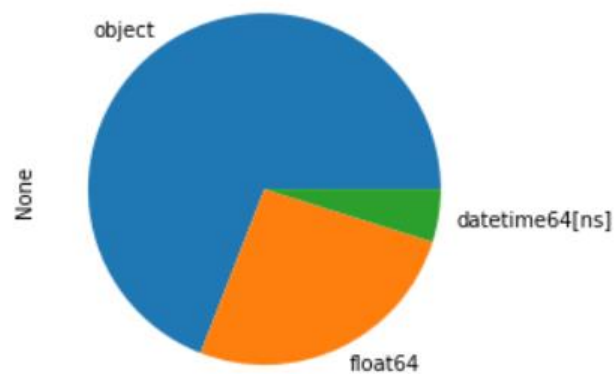


Abbildung 15: Kreisdiagramm für die MTART 1030

3.1.4 Vierter Datensatz

Dieser Datensatz setzt sich aus Feldern zusammen, die nur aus der MTART 1040 des Konzerns stammen. Es wird für die explorative Datenanalyse von einem Datensatz mit 20.000 Zeilen und 129 Feldern ausgegangen. Die Abbildung 16 stellt das Kreisdiagramm der Typen von Feldern und ihre Anteile für diesen Datensatz dar.

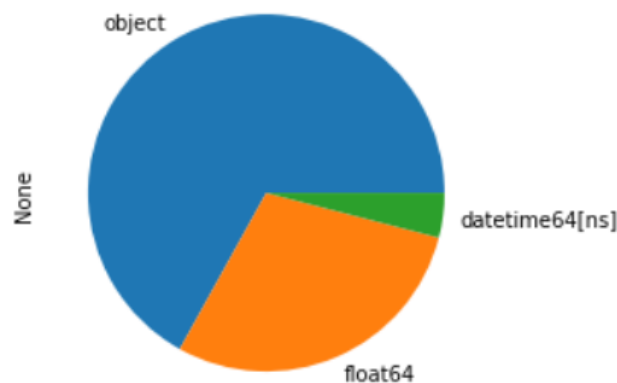


Abbildung 16 : Kreisdiagramm für die MTART 1040

3.2 Identifizierung von fehlenden Werten

Um einen Überblick über die fehlenden Daten in den jeweiligen Datensätzen zu bekommen, wurde für die vier verschiedenen Datensätze eine Heatmap verwendet. Es konnte mit Hilfe dieser Heatmap gezeigt werden, dass die meisten Felder befüllt worden waren. Das heißt, es sind jetzt fast keine Felder mit einer großen Anzahl an fehlenden Daten vorhanden.

Die Korrelation zwischen den Feldern werden auch betrachtet und es kann festgestellt werden, dass Korrelationen zwar vorhanden, aber nicht so hoch sind.

Lediglich wenige Felder zeigen eine hohe Korrelation. Dies gilt für alle vier Datensätze – siehe Abbildungen 17 bis 20.

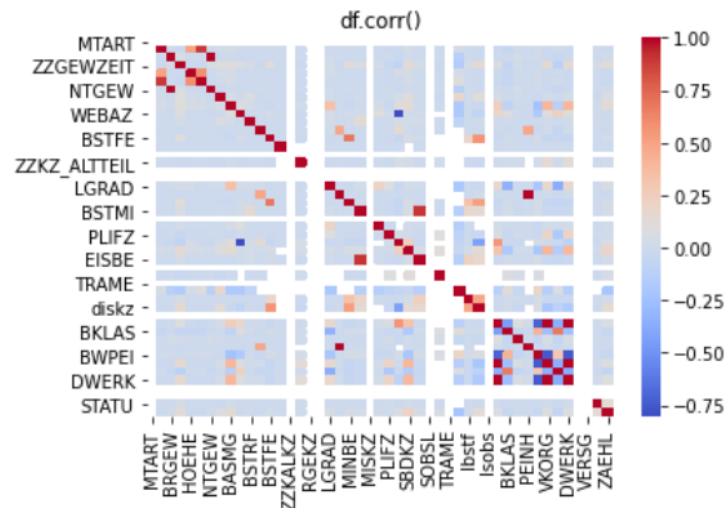


Abbildung 17 : Korrelationsmatrix der MTART 1010 vor der Datenvorverarbeitung

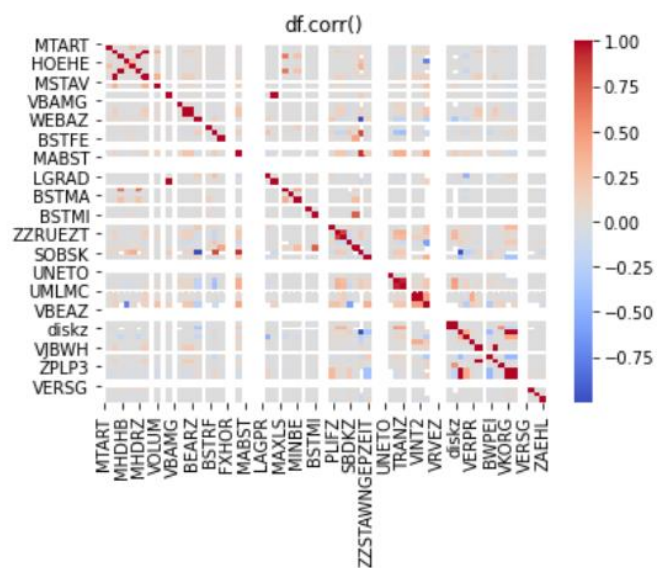


Abbildung 18 : Korrelationsmatrix der MTART 1000 vor der Datenvorverarbeitung

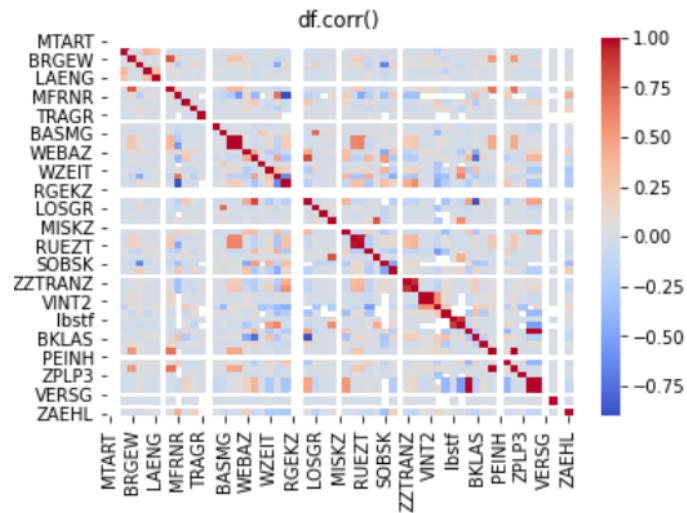


Abbildung 19 : Korrelationsmatrix der MTART 1030 vor der Datenvorverarbeitung

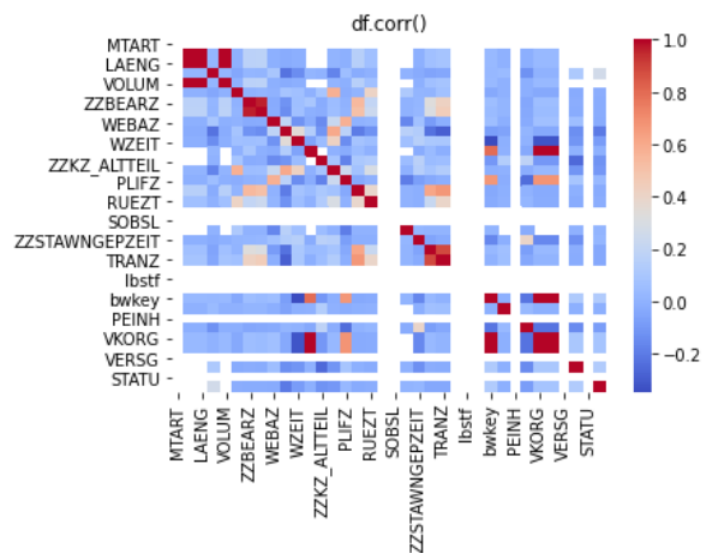


Abbildung 20 : Korrelationsmatrix der MTART 1040 vor der Datenvorverarbeitung

4 Datenvorverarbeitung

Die Datenvorverarbeitung ist ein wichtiger Schritt im Bereich des Data Minings, der Datenanalyse, -visualisierung und der Vorhersage in der Datenwissenschaft. Daten können verschiedene Formen annehmen, z. B. Text, Zahlen, Datumsformat, Bilder, Videos usw. Daher ist es obligatorisch, die Daten auf ein einheitliches Format vorzubereiten, um das Endergebnis zu erzielen. Die Datenvorverarbeitung ist ein Schritt, bei dem die Daten in ein einheitliches Format konvertiert oder transformiert werden (ICHI.PRO, 2020 - 2021).

Im Folgenden wird in Abschnitt 4.1 das Ziel der Datenvorverarbeitung erläutert, danach wird im Abschnitt 4.2 die Imputation von fehlenden Daten beschrieben. Im Abschnitt 4.3 wird die Kodierung von kategorischen Features erklärt, gefolgt vom Abschnitt 4.4, der die Normalisierung von numerischen Features erläutern wird. Im letzten Abschnitt 4.5 wird die Pipeline erklärt.

4.1 Ziel der Datenvorverarbeitung

Die Datenvorverarbeitung verfolgt das Ziel, Rohdaten mit Hilfe von Data-Mining-Techniken in ein verständliches Format für das Anwenden von ML- Algorithmen umzuwandeln.

Es wurden noch mehr Felder ausgeschlossen, darunter auch Felder mit hohen Korrelationen, weil sie redundante Informationen enthalten. Die Abbildungen 21 bis 24 zeigen die Heatmap von Korrelationsmatrizen für alle vier Datensätze, nachdem diese unbrauchbaren Felder ausgeschlossen wurden. An den Abbildungen kann erkannt werden, dass es keine Felder mehr gibt, die miteinander hoch korrelieren.

Out[272]:

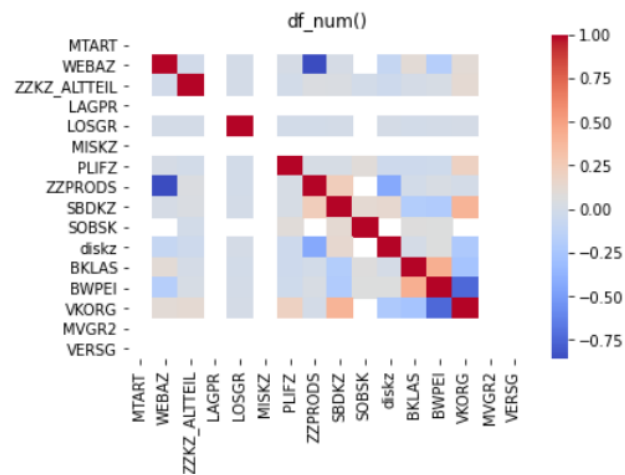


Abbildung 21 : Korrelationsmatrix der MTART 1010 nach der Datenvorverarbeitung

Out[186]:

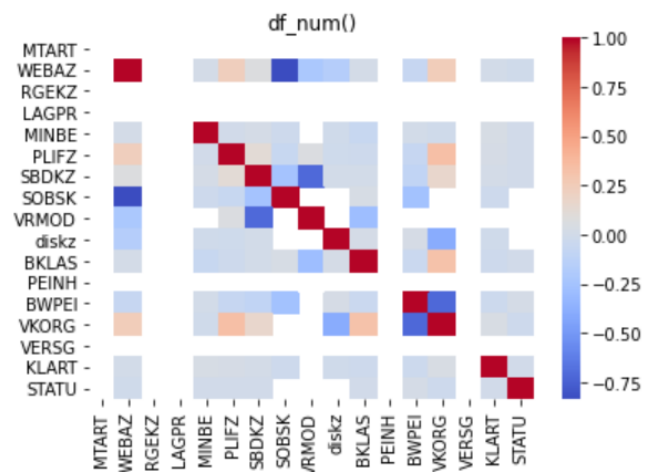


Abbildung 22 : Korrelationsmatrix der MTART 1000 nach der Datenvorverarbeitung

Out[456]:

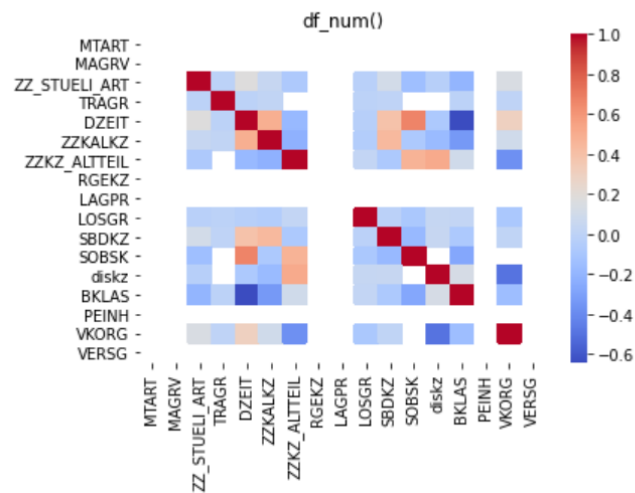


Abbildung 23 : Korrelationsmatrix der MTART 1030 nach der Datenvorverarbeitung

Out[164]:

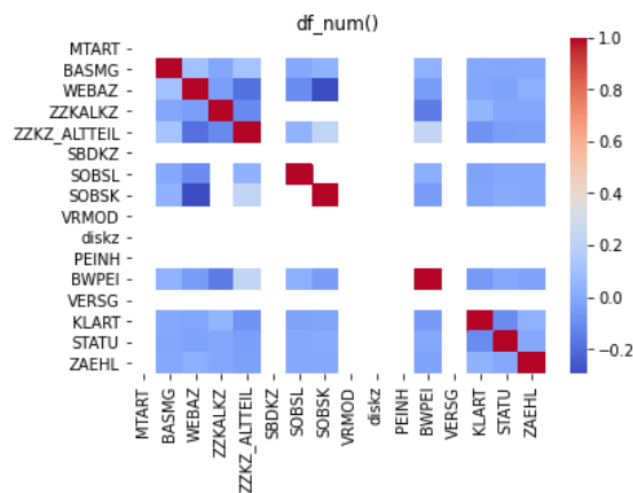


Abbildung 24 : Korrelationsmatrix der MTART 1040 nach der Datenvorverarbeitung

4.2 Imputation der fehlenden Daten

Der Unterschied zwischen den laborgenerierten Daten und den Daten in der realen Welt besteht darin, dass Daten in der realen Welt selten sauber und homogen sind. Insbesondere werden bei vielen interessanten Datensätzen einige Daten fehlen (VanderPlas, 2016, S. 119). Für die Arbeit mit fehlenden Daten können einerseits alle Zeilen mit fehlenden Daten aus dem Datensatz komplett gelöscht werden. Vor der Löschung der Zeile mit fehlenden Daten sollte sichergestellt werden, dass genügend Daten nach der Löschung zur Verfügung stehen werden. Die zweite Option ist, die fehlenden Daten mit einem statistischen Wert zu ersetzen, zum Beispiel dem Mittelwert oder dem Modus des Feldes. Für den Fall des Mittelwertes bedeutet es, dass jeder fehlende Wert durch den Mittelwert entlang jedes Feldes ersetzt wird. Diese Methode kann nur mit numerischen Daten verwendet werden. Für den Fall des Modus bedeutet es, dass jeder fehlende Wert durch den häufigsten vorkommenden Wert entlang jedes Feldes ersetzt wird. Diese Methode kann sowohl mit numerischen als auch mit kategorischen Feldern verwendet werden.

4.3 Kodierung von kategorischen Features

Oft werden Variable nicht als numerische Werte, sondern kategorisch angegeben. Diese Variablen müssen umkodiert werden, um sie für den Algorithmus verwenden zu können. Der häufigste verwendete Transformer ist One-Hot-Kodierung, Bei dieser Technik werden kategorische Features als ein numerisches Array kodiert. Die Eingabedaten für diesen Transformer sollte ein Array aus ganzen Zahlen oder Strings sein. Standardmäßig leitet der Encoder die Kategorien auf der Grundlage der einzelnen Werte in jedem Feature ab. Die Kodierung der kategorischen Features erzeugt eine binäre Spalte für jede Kategorie und gibt eine dünnbesetzte Matrix oder ein dichtes Array zurück (Scikit-learn developers, 2007 - 2020d).

4.4 Normalisierung von Numerischen Features

In der Datenwissenschaft ist es sehr wichtig, Daten zu normalisieren. Das heißt, sie alle auf die gleiche Skala zu transformieren, da dies das Lernen der Modelle von maschinellem Lernen erheblich erleichtert. Es gibt viele Normalisierungstechniken, aber es wird im Rahmen dieser Bachelorarbeit die Standardisierungstechnik verwendet. Sie standardisiert jede Variable so, dass ihr Mittelwert gleich Null und ihre Standardabweichung gleich Eins ist. Dafür wird das Z-Scoring verwendet. Dies ist die Berechnung dafür:

$$X_{scaled} = \frac{X - \mu_x}{\sigma_x}$$

Wobei:

μ_x der Mittelwert der Variable X ist.

σ_x die Standardabweichung der Variablen X ist.

X die Variable ist, die standardisiert wird.

X_{scaled} der standardisierte Wert der Variable X ist.

4.5 Pipeline von Features

Die Pipeline ist eine Transformationskette, die sich aus einem oder mehreren Transformern am Anfang der Kette und keinem oder mehreren Schätzern am Ende der Kette zusammensetzt. Die Pipeline wird auch zusammengesetzter Schätzer genannt. Die Tatsache, dass Transformer in einer Pipeline gruppiert sind, hat mehrere Vorteile:

- Eine Pipeline ist einfach zu bedienen,
- Sie vermeidet Datenlecks oder schlecht transformierte Daten,
- Sie ermöglicht die Kreuzvalidierung der gesamten Transformationskette im Falle des überwachten Lernens.

Dank Pipelines können Datensätze effizienter verarbeitet werden(Scikit-learn developers, 2007 - 2020b).

5 Untersuchung der Struktur der Daten

Da das Problem zuerst mittels unüberwachtes Lernen betrachtet wurde, musste die Struktur der Daten untersucht werden, um die richtige Anzahl von k (Ganzzahl, die angibt, in wie viele Gruppen der Datensatz unterteilt werden kann) für die Anwendung von dem Clustering Algorithmus herauszufinden. Im Abschnitt 5.1 wird die Ellbogen-Methode zur Bestimmung des richtigen Wertes von k mit Hilfe der Ellbogen-Methode erklärt.

5.1 Die Ellbogen-Methode

Die Ellbogen-Methode besteht darin, die Entwicklung der Kostenfunktion eines Modells in Abhängigkeit von der Anzahl der Cluster aufzuzeichnen und in einem Graphen eine Ellbogen-Zone zu bestimmen. Diese Ellbogen-Zone zeigt die Anzahl von Clustern an (O'REILLY, 2021).

Auf den Abbildungen 27 bis 30 ist ein Ellbogenbereich ersichtlich, der es ermöglicht, die Anzahl der Cluster zu bestimmen. Aus den Diagrammen kann angenommen werden, dass die Anzahl der Cluster zwischen 10 und 20 für den ersten und den zweiten Datensatz liegt oder zwischen 5 und 15 für den dritten und vierten Datensatz.

Out[296]:

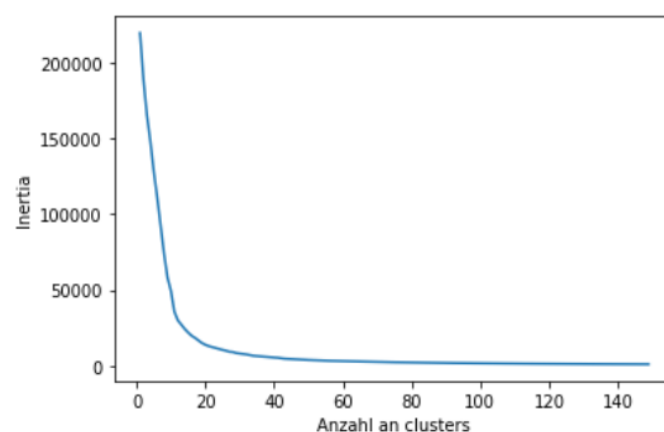


Abbildung 25: Ellbogen-Methode für die MTART 1010

Out[203]:

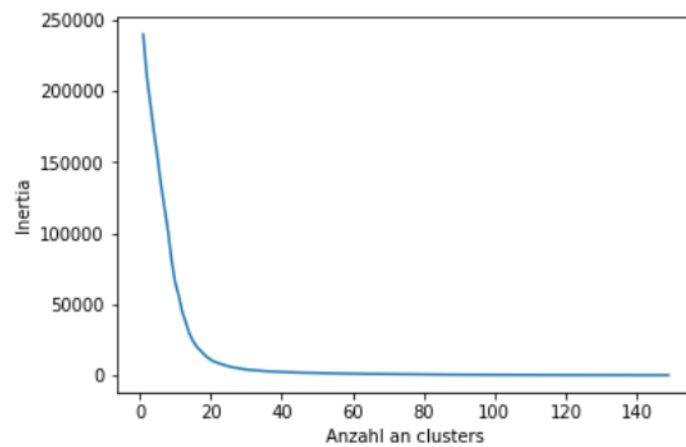


Abbildung 26 : Ellbogen-Methode für die MTART 1000

Out[228]:

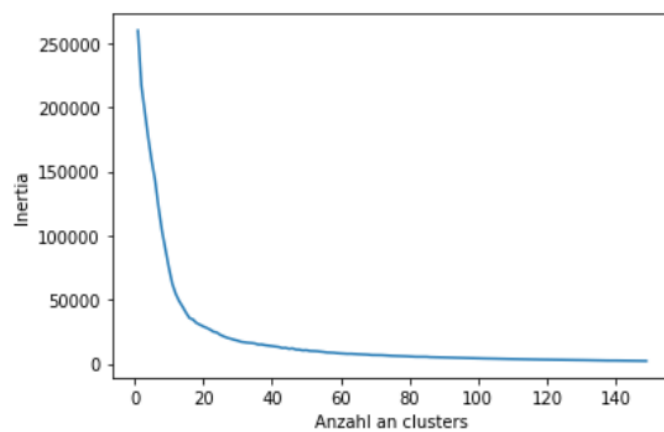


Abbildung 27 : Ellbogen-Methode für die MTART 1030

Out[180]:

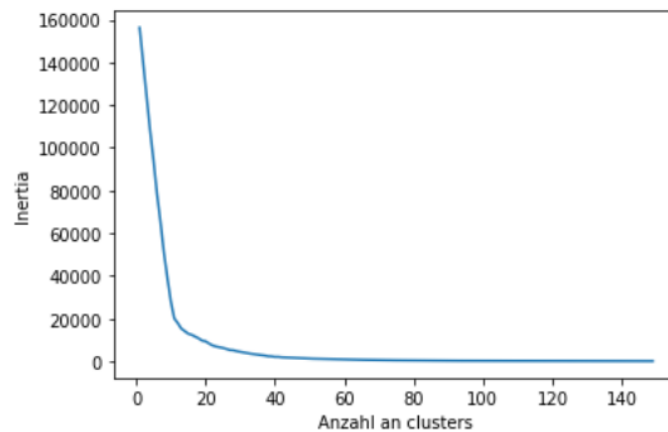


Abbildung 28 : Ellbogen-Methode für die MTART 1040

6 Modellierung

Die Aufgabe in diesem Abschnitt ist die Entwicklung eines Algorithmus von maschinellem Lernen. Nach der Bestimmung der Werte von k für die vier verschiedenen Datensätze wird der k-Means-Clustering-Algorithmus nur auf den numerischen Feldern angewendet, weil kein Abstandsbegriff in den kategorischen Feldern definiert werden kann und die Anwendung des k-Means-Clustering-Algorithmus sicher kein gutes Ergebnis auf diesen Feldern haben wird. Darüber hinaus ist der k-Means-Clustering-Algorithmus ein distanzbasierter Algorithmus, was bedeutet, dass er bessere Leistung auf numerischen Feldern haben wird, da die Abstandsbegriffe in den numerischen Feldern eine Rolle spielen.

Obwohl es eine Struktur in den Daten gibt, die es ermöglicht hat, die Daten in k -Cluster aufzuteilen, konnte nach der Verwendung des k-Means-Clustering-Algorithmus auf die vier Datensätze das Ziel der Aufgabe nicht erreicht werden.

Der Ansatz wird geändert und statt unüberwachten Lernen wird die Aufgabe mittels überwachten Lernens betrachtet. Im Abschnitt 6.1 wird die Überwachte Lernen-Methode zur Lösung der Aufgabe erklärt.

6.1 Anwendung vom überwachten Lernen

Nach der Anwendung des vorherigen k-Means-Clustering-Algorithmus konnte die Aufgabe nicht erledigt werden. Aus diesem Grund wurde beschlossen, den Ansatz zu ändern. In der Tat handelt es sich um eine Aufgabe mit Features (dies sind die Felder, die zur Vorhersage anderer Felder verwendet werden), die festgelegt werden müssen, und mit mehreren Targets, die jeweils vorhergesagt werden.

Es handelt sich um überwachtes Lernen, und zwar eine Klassifizierungsaufgabe. Für die Anwendung des ML-Algorithmus müssen die Features und die Label definiert werden.

Um Features und Labels auszuwählen, wird der Fokus auf kategoriale Felder gelegt, weil diese Felder für eine Klassifizierungsaufgabe besser geeignet sind.

Für die Features-Felder ist es sehr wichtig, dass diese Felder eine sehr große Anzahl von Klassen oder Kategorien haben, damit das Modell von maschinellem Lernen die maximal möglichen Informationen zur Vorhersage des Targets hat.

Es werden zwei ML-Algorithmen (der Complement Naive Bayes-Algorithmus und der Random Forest Classifier-Algorithmus) verwendet, um die verschiedenen Labels vorherzusagen und die Ergebnisse von den beiden Algorithmen miteinander zu vergleichen, damit es möglich wird, pro Target den besten Algorithmus auszuwählen.

6.1.1 Der Complement Naive Bayes Algorithmus (CNB)

Der Complement Naive Bayes Algorithmus (CNB) ist eine Anpassung des standardmäßigen multimedialen Naive Bayes Algorithmus (MNB), der sich besonders für unausgewogene Datensätze eignet (Scikit-learn developers, 2007 - 2020c).

Um besser zu verstehen, wie Bayes Algorithmen funktionieren, muss zunächst das Bayes-Theorem verstanden werden. Das Bayes-Theorem wird verwendet, um die Wahrscheinlichkeit eines Ereignisses zu ermitteln, wenn ein anderes Ereignis eingetreten ist (VanderPlas, 2016, S. 383). Die Formel des Bayes-Theorem ist:

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

wobei A und B Ereignisse sind. $P(A)$ ist die Wahrscheinlichkeit des Auftretens von A . $P(A/B)$ ist die Wahrscheinlichkeit des Auftretens von A unter der Voraussetzung, dass das Ereignis B bereits eingetreten ist. $P(B)$ die Wahrscheinlichkeit, dass das Ereignis B eintritt, kann nicht 0 sein, da es bereits eingetreten ist.

- Das Ereignis B wird auch als evidence (Beweis) bezeichnet.
- $P(A)$ wird auch als Priori von A bezeichnet (die Wahrscheinlichkeit des Ereignisses, bevor der Beweis gesehen wird).
- $P(A/B)$ wird auch als posteriori-Wahrscheinlichkeit von B bezeichnet, d.h. die Wahrscheinlichkeit des Ereignisses, nachdem der Beweis gesehen wurde.

Beim regulären Naive Bayes Algorithmus wird die Wahrscheinlichkeit einer gegebenen Menge von neuen Eingangsdaten für alle möglichen Klassen des Labels y berechnet. Danach wird die Klasse mit der größten Wahrscheinlichkeit ausgewählt. Die verschiedenen Naive-Bayes-Klassifikatoren unterscheiden sich hauptsächlich durch die Annahmen, die sie bezüglich der Verteilung der Daten treffen (VanderPlas, 2016, S. 383).

Es wird die Funktionsweise des CNB-Algorithmus im Kontext der Textverarbeitung erklärt, um sie besser zu verstehen. Doch diese

Funktionsweise gilt für alle Klassifizierungsaufgaben, in denen der CNB-Algorithmus zur Anwendung kommt.

Ein Dokument bzw. ein Text wird als eine Folge von Wörtern behandelt und es wird davon ausgegangen, dass jede Position eines Wortes unabhängig von jedem anderen erzeugt wird. Für die Klassifizierung wird angenommen, dass es eine feste Anzahl von Klassen, $c \in \{1, 2, \dots, m\}$ gibt, jede mit einer festen Reihe von Parametern. Bei einer Klassifikationsaufgabe sind die Anzahl der Klassen und Labels der Trainingsdaten für jede Klasse gegeben, nicht aber die Parameter für jede Klasse. Die Parameter müssen aus den Trainingsdaten geschätzt werden. CNB schätzt Parameter unter Verwendung von Daten aus allen Klassen außer der Klasse c (Rennie et al., 2003, S. 2).

Die CNB-Schätzungen sind bei unausgewogenen Daten effektiver, da sie eine gleichmäßigere Menge an Trainingsdaten pro Klasse verwenden, was die Klassifizierungsgenauigkeit des Modells verbessert (Rennie et al., 2003, S. 3).

Die Formel der Schätzungen von CNB lautet:

$$\hat{\theta}_{ci} = \frac{N_{\bar{c}i} + \alpha_i}{N_{\bar{c}} + \alpha}$$

wobei $N_{\bar{c}i}$ die Häufigkeit des Wortes i im Dokument, die in anderen Klassen als c vorkommen und $N_{\bar{c}}$ die Gesamtzahl von Worten ist, die in anderen Klassen als c vorkommen. α_i und α sind Parameter.

Die Berechnung der Gewichte für jede Schätzung der CNB ist:

$$\hat{w}_{\bar{c}i} = \log \hat{\theta}_{\bar{c}i}$$

und die Klassifizierungsregel lautet:

$$l_{CNB}(d) = \underset{c}{\operatorname{argmax}} \left[\log p(\hat{\theta}_c) - \sum_i f_i \log \frac{N_{\bar{c}i} + \alpha_i}{N_{\bar{c}} + \alpha} \right]$$

Wobei $\log p(\hat{\theta}_c)$ die Klassenprioritätsschätzung ist und f_i die Häufigkeit des Wortes i im Dokument d ist.

Das negative Vorzeichen steht für die Tatsache, dass es Dokumente gibt, die der Klasse c zuzuordnen sind, die schlecht mit den Komplement-Parameterschätzungen übereinstimmt (Rennie et al., 2003, S. 3). Das heißt die Klasse, die den kleinsten Wert aus der Klassifizierungsregel liefern wird.

Der Complement Naive Bayes ist nur die Umkehrung des regulären Naive Bayes. Bei der regulären Naive Bayes ist die Klasse mit dem größten Wahrscheinlichkeitswert, der sich aus der Klassifizierungsregel ergibt, die vorhergesagte Klasse. Dagegen ist die Klasse mit dem kleinsten Wahrscheinlichkeitswert, der sich aus der CNB- Klassifizierungsregel ergibt, die vorhergesagte Klasse.

Der CNB hat die Vorteile, dass er nicht viele Daten für sein Training benötigt und sehr schnell Ergebnisse liefert.

6.1.2 Der Random Forest Classifier Algorithmus (RFC)

Bevor der Random Forest Classifier Algorithmus vorgestellt wird, soll ein kurzer Überblick über Entscheidungsbäume (DT: Decision Trees) und insbesondere über den Algorithmus für Klassifizierungs- und Regressionsbäume (CART: Classification and Regression Trees) gegeben werden. Der CART-Algorithmus ist eine Methode zur Anwendung von DT durch rekursive und binäre Partitionierung des Trainingsdatensatzes. Die CART-Anwendung beginnt mit einem einzigen (Wurzel-)Knoten, der alle Trainingsdateninstanzen umfasst. CART generiert Entscheidungsknoten, die die Instanzen in immer reinere Partitionen entsprechend den Klassen ihrer Label aufteilen. Bei jeder Iteration werden für alle Knoten, Split-Kandidaten bewertet, die sich derzeit am Ende eines Entscheidungspfads befinden, aber noch nicht die Abbruchkriterien erfüllen. Der Split-Kandidat, der zwei untergeordnete Knoten mit der niedrigsten gewichteten Gesamt-Gini-Unreinheit I_G (Gini ist die Funktion zur Messung der Qualität eines Splits) erzeugt, wird immer ausgewählt, und diese untergeordneten Knoten werden dann dem wachsenden Baum hinzugefügt (Hatwell et al., 2020, S. 5754).

Die Gesamt-Gini-Unreinheit wird so berechnet:

$$I_G(Q) = \sum_{k=1}^K p_k (1 - p_k)$$

Wobei, p_k der Anteil der Instanzen mit dem Label y_k in einem Knoten Q und K die Anzahl der Klassen ist.

Um eine zuvor nicht gesehene Instanz zu klassifizieren, wird einfach dem Pfad dieser Instanz den Baum hinunter gefolgt, beginnend an der Wurzel und unter Beachtung der Aufteilungsbedingung jedes Entscheidungsknotens, bis ein Blattknoten (Knoten, der keine untergeordneten Knoten hat) erreicht ist (Hatwell et al., 2020, S. 5754).

Die Split-Bedingungen sind binäre Bedingungen, die für eine Instanz entweder wahr oder falsch sind. Jede Instanz kann also nur einem einzigen Pfad folgen und nur an einem einzigen Blattknoten ankommen.

Jeder Blattknoten deckt eine Teilmenge der Trainingsinstanzen ab und gibt die Mehrheitsklasse der Label dieser Instanzen als Ergebnis für die Klassifizierung zurück (Hatwell et al., 2020, S. 5754).

In der Abbildung 31 ist ein Beispiel eines Entscheidungsbaumes dargestellt.

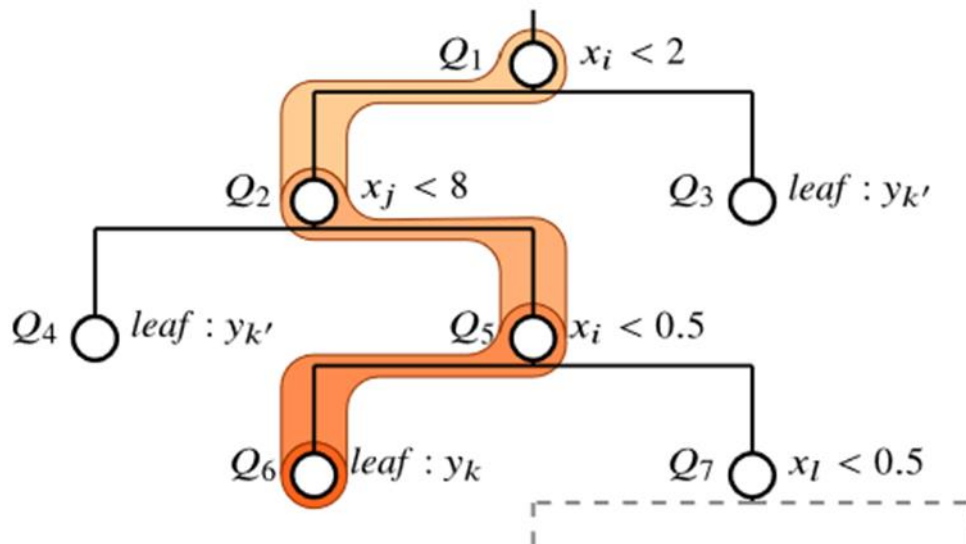


Abbildung 29: Darstellung eines Entscheidungsbaumes (Hatwell et al., 2020, S. 5755).

Um eine Instanz zu klassifizieren:

$\mathbf{x} = \{..., x_i = 0.1, x_j = 10...\}$,

Es wird bei Q_1 begonnen und den binären Aufteilungsbedingungen gefolgt, bis ein Blattknoten erreicht wird. In diesem Fall ist Q_6 der Blattknoten, der das Label y_k zurückgibt.

Ein RFC (Random Forest Classifier) ist ein Ensemble von DT-Basis Klassifikatoren, die zusammen parallel arbeiten und die Entscheidung über ihre Klassifizierung nach der Mehrheitswahl treffen. Dadurch wird die Klassifizierungsleistung eines Ensembles im Vergleich zu einem einzelnen Klassifikator verbessert. RFC-Modelle, die auf diese Weise konstruiert sind, sind in Bezug auf die Genauigkeit (ein statistisches Maß, das als Quotient aus den korrekten Vorhersagen eines Klassifikators und der Summe aller vom

Klassifikator gemachten Vorhersagen gebildet wird) mit den weit verbreiteten ML-Methoden konkurrenzfähig und sind außerdem robust gegenüber Überanpassung und unausgewogenen Daten (Hatwell et al., 2020, S. 5754–5755).

RFC hat aber den Nachteil, dass er sehr viel Zeit braucht, um seine Ergebnisse zu produzieren.

Die Leistung der ML-Algorithmen wird mit Hilfe von Gütemaßen bewertet. Die verwendeten Gütemaße für die Bewertung der ML-Algorithmen in dieser Arbeit werden in Abschnitt 6.2 vorgestellt.

6.2 Gütemaße für die Auswertung der Algorithmen

Ein Gütemaß ist ein Wert zur Bewertung der Leistung eines maschinellen Lernmodells. Die Tabellen 1 bis 10 präsentieren einen Vergleich der Durchschnittswerte der Gütemaße, die für die Label durch die verschiedenen Algorithmen erhalten wurden. Diese Gütemaße werden aus den Classification Report (Klassifizierungsberichten) jedes Labels entnommen, in dem nur die Durchschnittswerte dieser Gütemaße zur besseren Verdeutlichung berücksichtigt wurden.

Diese Gütemaßen sind: Die Genauigkeit, die Präzision und das Recall. Bevor erklärt wird, was es mit den Begriffen genau auf sich hat, müssen einige wichtige Begriffe bzw. Abkürzungen im Zusammenhang mit der Vorhersage einem Klassifizierungsalgorithmus erläutert werden, nämlich:

TN / True Negativ: Das Sample war negativ und wurde als negativ vom Modell vorhergesagt

TP / True Positiv: Das Sample war positiv und wurde als positiv vom Modell vorhergesagt

FN / Falsch Negativ: Das Sample war positiv, wurde aber als negativ vom Modell vorhergesagt

FP / Falsch Positiv: Das Sample war negativ, wurde aber als positiv vom Modell vorhergesagt

Die Genauigkeit ist ein statistisches Maß, das als Quotient aus den korrekten Vorhersagen eines Klassifikators und der Summe aller vom Klassifikator gemachten Vorhersagen gebildet wird (Grandini et al., 2020, S. 3).

$$\text{Genauigkeit} = \frac{TP + TN}{TP + FP + FN + TN}$$

Kurze Erläuterungen über Präzision, Recall, F1-Score:

- Die Präzision drückt den Anteil der Instanzen aus, die nach dem Modell positiv sind, und die tatsächlich positiv sind. Das heißt, Präzision sagt, wie sehr dem Modell vertraut werden kann, wenn es eine Instanz als positiv vorhersagt (Grandini et al., 2020, S. 2–3).

$$\text{Präzision} = TP / (TP + FP)$$

- Das Recall (Der Rückruf) sagt aus, wie viel Prozent der positiven Fälle identifiziert wurden. Das Recall misst die Fähigkeit des Modells, alle positiven Instanzen im Datensatz zu finden (Grandini et al., 2020, S. 3).

$$\text{Recall} = TP / (TP + FN)$$

- Das F1-Score sagt aus, wie viel Prozent der positiven Vorhersagen richtig waren. Er beschreibt das harmonische Mittel zwischen Recall und Precision und fasst damit die beiden Metriken zu einem Wert zusammen (Grandini et al., 2020, S. 5).

$$\text{F1-Score} = 2 * (\text{Recall} * \text{Präzision}) / (\text{Recall} + \text{Präzision})$$

Makro – Durchschnittspräzision (Precision macro avg): Ein arithmetischer Mittelwert der Präzisionswerte pro Klasse, dabei wird das Ungleichgewicht der Klassen nicht berücksichtigt (Grandini et al., 2020, S. 7).

$$\text{Makro – Durchschnittspräzision} = \frac{\sum_{k=1}^k \text{Precision}_k}{K}$$

K ist die gesamte Anzahl der Klasse.

Makro – Durchschnittsrecall (Recall macro avg): Ein arithmetischer Mittelwert der Recall-Werte pro Klasse, dabei wird das Ungleichgewicht der Klassen nicht berücksichtigt (Grandini et al., 2020, S. 7).

$$\text{Makro - Durchschnittsrecall} = \frac{\sum_{k=1}^k \text{Recall}_k}{K}$$

Makro - DurchschnittsF1-Score (F1_score macro avg): Ein arithmetischer Mittelwert der F1-Werte pro Klasse, dabei wird das Ungleichgewicht der Klassen nicht berücksichtigt (Grandini et al., 2020, S. 7).

$$\text{Makro - DurchschnittsF1 - Score} = 2 * \left(\frac{\frac{\sum_{k=1}^k \text{Recall}_k}{K} * \frac{\sum_{k=1}^k \text{Precision}_k}{K}}{\left(\frac{\sum_{k=1}^k \text{Recall}_k}{K}\right)^{-1} + \left(\frac{\sum_{k=1}^k \text{Precision}_k}{K}\right)^{-1}} \right)$$

Gewichtete - Durchschnittspräzision (Precision weighted avg): Der gewichtete Mittelwert der Präzisionswerte pro Klasse, dabei wird das Ungleichgewicht der Klassen berücksichtigt (Scikit-learn developers, 2007 - 2020a).

$$\text{Gewichtete - Durchschnittspräzision} = \frac{1}{\sum_{l \in L} |\hat{y}_l|} \sum_{l \in L} |\hat{y}_l| P(y_l, \hat{y}_l)$$

Wobei:

y die Menge (Sample, Label) der vorhergesagten Paare ist.

\hat{y} die Menge Sample, Label) der wahren Paare ist.

L die Menge der Labels ist.

y_l die Teilmenge von y mit Label l ist.

$P(y_l, \hat{y}_l)$ die Präzision der Klasse ist.

Gewichtetes – Durchschnittsrecall (Recall weighted avg): Der gewichtete Mittelwert der Recall-Werte pro Klasse, dabei wird das Ungleichgewicht der Klassen berücksichtigt (Scikit-learn developers, 2007 - 2020a).

$$\text{Gewichtetes – Durchschnittsrecall} = \frac{1}{\sum_{l \in L} |\hat{y}_l|} \sum_{l \in L} |\hat{y}_l| R(y_l, \hat{y}_l)$$

Wobei:

y die Menge (Sample, Label) der vorhergesagten Paare ist.

\hat{y} die Menge (Sample, Label) der wahren Paare ist.

L die Menge der Labels ist.

y_l die Teilmenge von y mit Label l ist.

$R(y_l, \hat{y}_l)$ das Recall der Klasse ist.

Gewichtetes – DurchschnittsF1Score (F1_score weighted avg): Der gewichtete Mittelwert der F1-Werte pro Klasse, dabei wird das Ungleichgewicht der Klassen berücksichtigt (Scikit-learn developers, 2007 - 2020a).

$$\text{Gewichtetes – DurchschnittsF1Score} = \frac{1}{\sum_{l \in L} |\hat{y}_l|} \sum_{l \in L} |\hat{y}_l| F_1(y_l, \hat{y}_l)$$

Wobei:

y die Menge (Sample, Label) der vorhergesagten Paare ist.

\hat{y} die Menge (Sample, Label) der wahren Paare ist.

L die Menge der Labels ist.

y_l die Teilmenge von y mit Label l ist.

$F_1(y_l, \hat{y}_l)$ der F1-Score der Klasse ist.

Da mit mehreren unausgewogenen Klassen gearbeitet wird, sind die gewichteten Gütemaße, wie die gewichtete Durchschnittspräzision, das gewichtete Durchschnittsrecall und das gewichtete DurchschnittsF1Score in Betracht zu

ziehen, weil sie die Unausgewogenheit in jeder Klasse berücksichtigen. Außerdem muss auch die Genauigkeit betrachtet werden.

7 Auswertung der Ergebnisse der vier Datensätze

Nachdem die ML-Algorithmen und die verschiedenen Gütemaße erklärt wurden, wird im Unterabschnitt 7.1 die Auswertung der Ergebnisse des ersten Datensatzes für die zwei verwendeten Algorithmen dargestellt. Im Unterabschnitt 7.2 wird die Auswertung der Ergebnisse des zweiten Datensatzes dargestellt, gefolgt vom Unterabschnitt 7.3, wo die Auswertungen der Ergebnisse des dritten Datensatzes dargestellt wird. Danach wird im letzten Unterabschnitt 7.4 die Auswertungen der Ergebnisse vom vierten Datensatz präsentiert.

7.1 Auswertung der Ergebnisse des ersten Datensatzes

Für diesen Datensatz wurden 12 Features verwendet und mit diesen Features 28 je als ein Target vorhergesagt. Verwendet wird einen Datensatz mit 30.000 Zeilen und 124 Feldern.

7.1.1 Auswertung der Ergebnisse aus dem RFC – Algorithmus

Nachdem die Features und Targets ausgewählt und die Algorithmen verwendet wurden, ergeben sich in der Tabelle 1 die Auswertungen aus dem RFC-Algorithmus. Diese 7 Targets (VOLEH, ZZVERSCHLKZA, MEABM, GEWEI, DISLS, DWERK, ZAEHL) haben folgende Gütemaße:

Das Target VOLEH hat ein Gewichtetes – Durchschnittsrecall von 75% und eine Genauigkeit von 75%.

Das Target ZZVERSCHLKZA hat eine Gewichtete – Durchschnittspräzision von 69%, ein Gewichtetes–Durchschnittsrecall von 52%, ein Gewichtetes – DurchschnittsF1Score 53% und eine Genauigkeit von 52%.

Das Target MEABM hat ein Gewichtetes – Durchschnittsrecall von 77% und eine Genauigkeit von 71%.

Das Target GEWEI hat ein Gewichtetes – Durchschnittsrecall von 58%, ein Gewichtetes – DurchschnittsF1Score von 67% und eine Genauigkeit von 58%.

Das Target DISLS hat ein Gewichtetes – Durchschnittsrecall von 72%, ein Gewichtetes – DurchschnittsF1Score von 76% und eine Genauigkeit von 72%.

Das Target DWERK hat ein Gewichtetes – Durchschnittsrecall von 55%, ein Gewichtetes – DurchschnittsF1Score von 63% und eine Genauigkeit von 55%.

Das Target ZAEHL hat ein Gewichtetes – Durchschnittspräzision von 62%, ein Gewichtetes – Durchschnittsrecall von 48%, ein Gewichtetes – DurchschnittsF1Score von 53% und eine Genauigkeit von 48%.

Für diese 7 Felder kann festgestellt werden, dass es mindestens einen Wert für die betrachtete Gütemaße gibt, der unter 80% liegt.

Die restlichen 21 Target (MEINS , OCM PF, KOSGR, VTWEG, SPART, EKWSL, DISMM, FHORI, STRGR, VINT1, VINT2, HRKFT, LGRAD, diskz, LOSGR, SBDKZ, ZZKZ_ALTTEIL, BWPEI, VKORG, BASMG, STATU) haben für die betrachteten Gütemaße in der Tabelle 1 Werte, die zwischen 80 und 100% liegen.

Tabelle 1: Ergebnisse des RFC des ersten Datensatzes

MTART_1010										
Features	ZZAUSLO ESER	MFRNR	MATKL	WERKS	DISPO	PSTAT				
Features	ZZTEXTN	MTART	BKLAS	WEBAZ	PLIFZ	ZZPRO DS				
Algorithmus										
RFC - Algorithmus										
Target	MEINS	OCMPF	KOSGR	VTWEG	VOLEH	SPART	ZZVERSC HLKZA	EKWSL	MEABM	GEWEI
Gütemaße										
Precision (macro avg)	0.17	0.95	0.95	0.99	0.47	0.73	0.44	0.37	0.38	0.38
Precision (weighted avg)	0.97	0.96	0.98	0.98	0.90	0.82	0.69	0.80	0.90	0.92
Recall (macro avg)	0.64	0.95	0.98	0.92	0.77	0.92	0.56	0.82	0.81	0.48
Recall (weighted avg)	0.65	0.96	0.98	0.98	0.75	0.81	0.52	0.68	0.71	0.58
F1_score (macro avg)	0.19	0.95	0.97	0.95	0.51	0.73	0.43	0.39	0.43	0.32
F1_score (weighted avg)	0.77	0.96	0.98	0.98	0.80	0.81	0.53	0.72	0.77	0.67
accuracy	0.65	0.96	0.98	0.98	0.75	0.81	0.52	0.68	0.71	0.58
Target	DISLS	DISMM	FHORI	STRGR	VINT1	VINT2	HRKFT	LGRAD		

Gütemaßen										
Precision (macro avg)	0.53	0.29	0.48	0.58	0.29	0.28	0.32	0.50		
Precision (weighted avg)	0.84	0.98	0.97	0.88	0.99	0.99	0.83	1.00		
Recall (macro avg)	0.68	0.77	0.83	0.85	0.95	0.82	0.63	0.50		
Recall (weighted avg)	0.72	0.87	0.87	0.69	0.81	0.82	0.68	0.99		
F1_score (macro avg)	0.55	0.35	0.54	0.59	0.34	0.34	0.34	0.50		
F1_score (weighted avg)	0.76	0.92	0.91	0.75	0.88	0.89	0.73	1.00		
accuracy	0.72	0.87	0.87	0.69	0.81	0.82	0.68	0.99		
Target	diskz	LOSGR	SBDKZ	ZZKZ_AL TTEIL	BWPEI	VKORG	DWERK	BASMG	STATU	ZAHL
Gütemaße										
Precision (macro avg)	0.74	0.39	0.91	0.52	0.72	1.00	0.83	0.46	0.55	0.19
Precision (weighted avg)	0.99	0.95	0.90	0.98	0.75	1.00	0.94	0.92	1.00	0.62
Recall (macro avg)	0.92	0.72	0.88	0.67	0.69	1.00	0.92	0.57	0.85	0.61
Recall (weighted avg)	0.98	0.91	0.90	0.85	0.65	1.00	0.55	0.83	0.98	0.48
F1_score (macro avg)	0.80	0.43	0.89	0.50	0.65	1.00	0.83	0.48	0.59	0.20
F1_score (weighted avg)	0.98	0.92	0.89	0.91	0.64	1.00	0.63	0.86	0.99	0.53
accuracy	0.98	0.91	0.90	0.85	0.65	1.00	0.55	0.83	0.98	0.48

7.1.2 Auswertung der Ergebnisse aus dem CNB – Algorithmus

Nachdem die Features und Targets ausgewählt und die Algorithmen verwendet wurden, ergeben sich in der Tabelle 2 die Auswertungen aus dem CNB – Algorithmus. Im Gegensatz zum RFC-Algorithmus gibt mit dem CNB-Algorithmus kein Target, wo das Modell für die betrachteten Gütemaße Werte unter 80% hat. Das heißt, dass die Werte der betrachteten Gütemaße für diese Target mindestens zwischen 80 und 100% liegen.

Tabelle 2: Ergebnisse des CNB des ersten Datensatzes

MTART_1010										
Features Felder	ZZAUSLO ESER	MFRNR	MATKL	WERKS	DISPO	PSTAT				
Features Felder	ZZTEXTN	MTART	BKLAS	WEBAZ	PLIFZ	ZZPRO DS				
Algorithmus										
CNB - Algorithmus										
Target	MEINS	OCPMF	KOSGR	VTWEG	VOLEH	SPART	ZZVERSC HLKZA	EKWSL	MEABM	GEWEI
Gütemaße										
Precision (macro avg)	0.31	0.99	0.99	0.99	0.80	0.47	0.74	0.76	0.60	0.61

Precision (weighted avg)	0.98	0.99	0.99	0.98	0.94	0.90	0.92	0.88	0.94	0.96
Recall (macro avg)	0.30	0.99	0.97	0.94	0.76	0.56	0.94	0.68	0.52	0.55
Recall (weighted avg)	0.98	0.99	0.99	0.98	0.94	0.90	0.91	0.88	0.94	0.96
F1_score (macro avg)	0.30	0.99	0.98	0.96	0.77	0.50	0.78	0.71	0.54	0.57
F1_score ((weighted avg)	0.98	0.99	0.99	0.98	0.94	0.90	0.92	0.88	0.94	0.96
accuracy	0.9–8	0.99	0.99	0.98	0.94	0.90	0.91	0.88	0.94	0.96
Target	DISLS	DISMM	FHORI	STRGR	VINT1	VINT2	HRKFT	LGRAD		
Gütemaße										
Precision (macro avg)	0.73	0.72	0.85	0.90	0.80	0.80	0.57	0.50		
Precision (weighted avg)	0.93	0.99	1.00	0.91	0.99	0.99	0.85	1.00		
Recall (macro avg)	0.73	0.74	0.78	0.76	0.40	0.40	0.46	0.50		
Recall (weighted avg)	0.93	0.98	1.00	0.89	0.99	0.99	0.86	1.00		
F1_score (macro avg)	0.73	0.69	0.81	0.78	0.48	0.48	0.48	0.50		
F1_score (weighted avg)	0.93	0.98	1.00	0.89	0.99	0.99	0.84	1.00		
accuracy	0.93	0.98	1.00	0.89	0.99	0.99	0.86	1.00		
Target	diskz	LOSGR	SBDKZ	ZZKZ_AL TTEIL	BWPEI	VKORG	DWERK	BASMG	STATU	ZAEHL
Gütemaße										
Precision (macro avg)	1.00	0.66	0.94	0.49	0.81	1.00	0.94	0.86	0.50	0.19
Precision (weighted avg)	0.99	1.00	0.94	0.98	0.82	1.00	0.95	0.93	1.00	0.63
Recall (macro avg)	0.75	0.66	0.94	0.50	0.81	1.00	0.88	0.78	0.50	0.20
Recall (weighted avg)	0.99	1.00	0.94	0.99	0.80	1.00	0.95	0.94	1.00	0.63
F1_score (macro avg)	0.83	0.66	0.94	0.50	0.80	1.00	0.89	0.82	0.50	0.19
F1_score ((weighted avg)	0.99	1.00	0.94	0.98	0.80	1.00	0.93	0.93	1.00	0.62
accuracy	0.99	1.00	0.94	0.99	0.80	1.00	0.95	0.94	1.00	0.63

7.2 Auswertung der Ergebnisse des zweitens Datensatzes

Für diesen Datensatz wurden 12 Features verwendet und mit diesen Features 29 je als ein Target vorhergesagt. Verwendet wird einen Datensatz mit 30.000 Zeilen und 184 Feldern.

7.2.1 Auswertung der Ergebnisse aus dem RFC – Algorithmus

Nachdem die Features und Targets ausgewählt und die Algorithmen verwendet wurden, ergeben sich in der Tabelle 4 die Auswertungen aus dem RFC-Algorithmus.

Die Tabelle 3 ist aus der Tabelle 4 abgeleitet und zeigt die Tagets, welche mindestens einen Wert von unter 80% für die betrachtete Gütemaße ergeben.

Die restlichen 15 Target (MEINS, OCMPPF, KOSGR, MTVFP, HRKFT, MAGRV, SSQSS, HERKL, AWSLS, SBDKZ, BSTFE, BWPEI, MINBE, VKORG, LGRAD) haben für die betrachteten Gütemaße in der Tabelle 4 Werte, die mindestens zwischen 80 und 100% liegen.

Tabelle 3: Targets, deren Wert für die betrachtete Gütemaße mindestens unter 80% liegt.

MTART_1000							
Features Felder	ZZAUSLOESE R	MFRN R	MATKL	WERK S	DISPO	PSTAT	
Features Felder	ZZTEXTN	MTART	BKLAS	WEBA Z	PLIFZ	VRMOD	
Algorithmus							
RFC - Algorithmus							
Zielfelder	DISMM	FHORI	MSTAE	VOLEH	SPART	ZZVERSCHLKZ A	EKWS L
Gütemaßen							
Precision (macro avg)	0.29	0.48	0.21	0.31	0.36	0.41	0.31
Precision (weighted avg)	1.00	0.97	0.97	0.62	0.86	0.85	0.90
Recall (macro avg)	0.97	0.76	0.63	0.65	0.72	0.67	0.56
Recall (weighted avg)	0.77	0.69	0.48	0.47	0.26	0.59	0.79
F1_score (macro avg)	0.33	0.49	0.19	0.30	0.37	0.40	0.33

F1_score (weighted avg)	0.87	0.79	0.63	0.50	0.29	0.67	0.82
accuracy	0.77	0.69	0.48	0.47	0.26	0.59	0.79
Target	MMSTA	DWER K	BASM G	ZAEHL	MEAB M	GEWEI	DISLS
Gütemaßen							
Precision (macro avg)	0.18	0.72	0.09	0.10	0.46	0.40	0.51
Precision (weighted avg)	0.94	0.99	0.84	0.64	0.67	0.90	0.79
Recall (macro avg)	0.54	0.88	0.16	0.20	0.62	0.50	0.73
Recall (weighted avg)	0.31	0.78	0.24	0.35	0.57	0.77	0.73
F1_score (macro avg)	0.16	0.72	0.06	0.08	0.44	0.40	0.51
F1_score (weighted avg)	0.43	0.87	0.33	0.44	0.61	0.82	0.75
accuracy	0.31	0.78	0.24	0.35	0.57	0.77	0.73

Tabelle 4: Ergebnisse des RFC des zweiten Datensatzes

MTART_1000										
Features Felder	ZZAUSLO ESER	MFRNR	MATKL	WERKS	DISPO	PSTAT				
Features Felder	ZZTEXTN	MTART	BKLAS	WEBAZ	PLIFZ	VRMOD				
Algorithmus										
RFC - Algorithmus										
Target	MEINS	OCMPF	KOSGR	VOLEH	SPART	ZZVERSC HLKZA	EKWSL	MEABM	GEWEI	DISLS
Gütemaße										
Precision (macro avg)	0.33	0.99	0.89	0.31	0.36	0.41	0.31	0.46	0.40	0.51
Precision (weighted avg)	1.00	1.00	0.99	0.62	0.86	0.85	0.90	0.67	0.90	0.79
Recall (macro avg)	1.00	0.99	0.98	0.65	0.72	0.67	0.56	0.62	0.50	0.73
Recall (weighted avg)	0.99	1.00	0.99	0.47	0.26	0.59	0.79	0.57	0.77	0.73
F1_score (macro avg)	0.39	0.99	0.93	0.30	0.37	0.40	0.33	0.44	0.40	0.51
F1_score (weighted avg)	0.99	1.00	0.99	0.50	0.29	0.67	0.82	0.61	0.82	0.75
accuracy	0.99	1.00	0.99	0.47	0.26	0.59	0.79	0.57	0.77	0.73
Target	DISMM	FHORI	MTVFP	HRKFT	MSTAE	MAGRV	SSQSS	HERKL	MMSTA	AWLS
Gütemaße										
Precision (macro avg)	0.29	0.48	0.70	0.50	0.21	0.48	0.36	0.27	0.18	0.67
Precision (weighted avg)	1.00	0.97	0.97	0.98	0.97	1.00	0.98	0.93	0.94	1.00
Recall (macro avg)	0.97	0.76	0.99	0.95	0.63	0.66	0.88	0.59	0.54	0.99
Recall (weighted avg)	0.77	0.69	0.92	0.88	0.48	0.97	0.80	0.80	0.31	0.98
F1_score (macro avg)	0.33	0.49	0.76	0.55	0.19	0.53	0.41	0.28	0.16	0.68
F1_score (weighted avg)	0.87	0.79	0.94	0.92	0.63	0.99	0.87	0.83	0.43	0.99
accuracy	0.77	0.69	0.92	0.88	0.48	0.97	0.80	0.80	0.31	0.98

Target	SBDKZ	BSTFE	BWPEI	MINBE	VKORG	DWERK	BASMG	ZAEHL	LGRAD	
Gütemaße										
Precision (macro avg)	0.88	0.12	0.79	0.37	0.93	0.72	0.09	0.10	0.52	
Precision (weighted avg)	0.91	1.00	0.79	1.00	1.00	0.99	0.84	0.64	1.00	
Recall (macro avg)	0.91	0.41	0.68	0.54	0.93	0.88	0.16	0.20	0.74	
Recall (weighted avg)	0.90	0.90	0.68	0.95	1.00	0.78	0.24	0.35	0.97	
F1_score (macro avg)	0.89	0.15	0.65	0.40	0.93	0.72	0.06	0.08	0.58	
F1_score (weighted avg)	0.91	0.95	0.65	0.97	1.00	0.87	0.33	0.44	0.98	
accuracy	0.90	0.90	0.68	0.95	1.00	0.78	0.24	0.35	0.97	

7.2.2 Auswertung der Ergebnisse aus dem CNB – Algorithmus

Nachdem die Features und Target ausgewählt und die Algorithmen verwendet werden, ergeben sich in der Tabelle 5 die Auswertung aus dem CNB-Algorithmus. Im Gegensatz zum RFC – Algorithmus gibt es mit dem CNB-Algorithmus nur 5 Targets (VOLEH, MEABM, BWPEI, BASMG, ZAEHL.), deren Mindestwert für die betrachtete Gütemaße unter 80% liegt. Das Target VOLEH hat Gewichtete-Durchschnittspräzision von 76%, ein das Gewichtetes–Durchschnittsrecall von 75%, ein Gewichtetes– DurchshnittsF1Score von 75% und eine Genauigkeit von 75%.

Das Target MEABM hat eine Gewichtete-Durchschnittspräzision 74%, ein Gewichtetes–Durchschnittsrecall von 75%, ein Gewichtetes–DurchshnittsF1Score von 75% und eine Genauigkeit von 75%.

Das Target BWPEI hat eine Gewichtete-Durchschnittspräzision 79%, ein Gewichtetes – Durchschnittsrecall von 70%, ein Gewichtetes – DurchshnittsF1Score von 68% und eine Genauigkeit von 70%.

Das Target BASMG hat eine Gewichtete-Durchschnittspräzision 73% und ein Gewichtetes – DurchshnittsF1Score von 74%.

Das Target ZAEHL hat eine Gewichtete-Durchschnittspräzision 53%, ein Gewichtetes–Durchschnittsrecall von 65%, ein Gewichtetes–DurchshnittsF1Score von 56% und eine Genauigkeit von 65%.

Die restlichen 24 Targets (MEINS, OCMPPF, KOSGR, SPART, ZZVERSCHLKZA, EKWSL, GEWEI, DISLS, DISMM, FHORI, MTVFP, HRKFT, MSTAE, MAGRV,

SSQSS, HERKL, MMSTA, AWSLS, SBDKZ, BSTFE, MINBE, VKORG, DWERK, LGRAD) haben für die betrachteten Gütemaße in der Tabelle 5 Werte, die zwischen 80 und 100% liegen.

Tabelle 5: Ergebnisse des CNB des zweiten Datensatzes

MTART_1000										
Features Felder	ZZAUSLO ESER	MFRNR	MATKL	WERKS	DISPO	PSTAT				
Features Felder	ZZTEXTN	MTART	BKLAS	WEBAZ	PLIFZ	VRMOD				
Algorithmus										
CNB - Algorithmus										
Zielfelder	MEINS	OCMPF	KOSGR	VOLEH	SPART	ZZVERSCH LKZA	EKWSL	MEABM	GEWEI	DISLS
Gütemaße										
Precision (macro avg)	0.25	0.99	0.97	0.46	0.68	0.81	0.56	0.65	0.58	0.80
Precision (weighted avg)	1.00	1.00	1.00	0.76	0.90	0.91	0.93	0.74	0.93	0.85
Recall (macro avg)	0.25	1.00	0.99	0.56	0.56	0.64	0.39	0.69	0.44	0.78
Recall (weighted avg)	1.00	1.00	1.00	0.75	0.91	0.91	0.93	0.75	0.93	0.85
F1_score (macro avg)	0.25	1.00	0.98	0.48	0.59	0.67	0.39	0.66	0.47	0.79
F1_score (weighted avg)	1.00	1.00	1.00	0.75	0.89	0.90	0.92	0.75	0.92	0.85
accuracy	1.00	1.00	1.00	0.75	0.91	0.91	0.93	0.75	0.93	0.85
Zielfelder	DISMM	FHORI	MTVFP	HRKFT	MSTAE	MAGRV	SSQSS	HERKL	MMSTA	AWSLS
Gütemaße										
Precision (macro avg)	0.29	0.55	0.86	0.32	0.45	0.33	0.87	0.33	0.53	0.66
Precision (weighted avg)	0.99	0.95	1.00	0.96	0.97	1.00	0.99	0.90	0.96	1.00
Recall (macro avg)	0.29	0.51	0.85	0.33	0.26	0.33	0.74	0.28	0.29	0.67
Recall (weighted avg)	1.00	0.97	1.00	0.98	0.98	1.00	0.99	0.90	0.97	1.00
F1_score (macro avg)	0.29	0.50	0.85	0.33	0.28	0.33	0.80	0.28	0.32	0.67
F1_score (weighted avg)	0.99	0.96	1.00	0.97	0.98	1.00	0.99	0.89	0.96	1.00
accuracy	1.00	0.97	1.00	0.98	0.98	1.00	0.99	0.90	0.97	1.00
Target	SBDKZ	BSTFE	BWPEI	MINBE	VKORG	DWERK	BASMG	ZAEHL	LGRAD	
Gütemaße										
Precision (macro avg)	0.97	0.10	0.79	0.17	0.91	0.73	0.15	0.12	0.67	
Precision (weighted avg)	0.97	0.99	0.79	1.00	1.00	0.99	0.73	0.53	1.00	
Recall (macro avg)	0.97	0.10	0.70	0.17	0.88	0.70	0.07	0.12	0.67	
Recall (weighted avg)	0.97	1.00	0.70	1.00	1.00	1.00	0.80	0.65	1.00	
F1_score (macro avg)	0.97	0.10	0.67	0.17	0.89	0.71	0.09	0.11	0.67	
F1_score (weighted avg)	0.97	0.99	0.68	1.00	1.00	0.99	0.74	0.56	1.00	

accuracy	0.97	1.00	0.70	1.00	1.00	1.00	0.80	0.65	1.00	
----------	------	------	------	------	------	------	------	------	------	--

7.3 Auswertung der Ergebnisse des dritten Datensatzes

Für diesen Datensatz wurden 12 Features verwendet und mit diesen Features 29 je als ein Target vorhergesagt. Verwendet wird einen Datensatz mit 30.000 Zeilen und 165 Feldern.

7.3.1 Auswertung der Ergebnisse aus dem RFC – Algorithmus

Nachdem die Features und Targets ausgewählt und die Algorithmen verwendet wurden, ergeben sich in der Tabelle 7 die Auswertungen aus dem RFC – Algorithmus.

Die Tabelle 6 ist aus der Tabelle 7 abgeleitet und zeigt die Targets, deren Mindestwert für die betrachtete Gütemaße unter 80% liegt.

Die restlichen 18 (MEINS, KOSGR, VTWEG, OCMPF, GEWEI, DISMM, MTVFP, HRKFT, SSQSS, MTPOS, HERKL, SBDKZ, ZZKZ_ALTTEIL, SOBSK, TRAGR, VKORG, DWERK, LGRAD) haben für die betrachteten Gütemaße in der Tabelle 7 Werte, die zwischen 80 und 100% liegen.

Tabelle 6: Target, deren Wert für die betrachtete Gütemaße mindestens unter 80% liegt

MTART_1030						
Features Felder	EKGRP	PRCTR	MATKL	WERKS	DISPO	PSTAT
Features Felder	LABOR	MTART	BKLAS	ZZ_STUELI_AR T	DZEIT	WZEIT
Algorithmus						
RFC - Algorithmus						
Target	ZZVERSCHKZA	EKWSL	MEABM	VOLEH	SPART	MSTAE
Gütemaßen						
Precision (macro avg)	0.55	0.34	0.40	0.49	0.32	0.17
Precision (weighted avg)	0.82	0.90	0.79	0.80	0.92	0.95
Recall (macro avg)	0.77	0.70	0.60	0.58	0.62	0.50
Recall (weighted avg)	0.77	0.67	0.58	0.53	0.58	0.19
F1_score (macro avg)	0.59	0.35	0.37	0.36	0.30	0.12
F1_score (weighted avg)	0.78	0.74	0.66	0.61	0.65	0.28

accuracy	0.77	0.67	0.58	0.53	0.58	0.19
Target	DISLS	ZZKALK Z	STRGR	BWPEI	ZAEHL	
Gütemaßen						
Precision (macro avg)	0.52	0.64	0.51	0.76	0.23	
Precision (weighted avg)	0.84	0.78	0.88	0.85	0.69	
Recall (macro avg)	0.82	0.57	0.87	0.79	0.30	
Recall (weighted avg)	0.77	0.76	0.74	0.72	0.54	
F1_score (macro avg)	0.58	0.59	0.54	0.72	0.22	
F1_score (weighted avg)	0.78	0.75	0.78	0.73	0.60	
accuracy	0.77	0.76	0.74	0.72	0.54	

Tabelle 7: Ergebnisse des RFC des dritten Datensatzes

MTART_1030										
Features Felder	EKGRP	PRCTR	MATKL	WERKS	DISPO	PSTAT				
Features Felder	LABOR	MTART	BKLAS	ZZ_STUEL L_ART	DZEIT	WZEIT				
Algorithmus										
RFC - Algorithmus										
Target	MEINS	KOSGR	VTWEG	VOLEH	SPART	OCMPF	ZZVERSC HLKZA	EKWSL	MEABM	GEWEI
Gütemaße										
Precision (macro avg)	0.25	1.00	0.98	0.49	0.32	0.98	0.55	0.34	0.40	0.35
Precision (weighted avg)	1.00	0.99	0.97	0.80	0.92	0.99	0.82	0.90	0.79	0.99
Recall (macro avg)	0.49	0.99	0.92	0.58	0.62	0.96	0.77	0.70	0.60	0.57
Recall (weighted avg)	0.94	0.99	0.97	0.53	0.58	0.99	0.77	0.67	0.58	0.91
F1_score (macro avg)	0.25	0.99	0.95	0.36	0.30	0.97	0.59	0.35	0.37	0.35
F1_score (weighted avg)	0.97	0.99	0.97	0.61	0.65	0.99	0.78	0.74	0.66	0.95
accuracy	0.94	0.99	0.97	0.53	0.58	0.99	0.77	0.67	0.58	0.91
Target	DISLS	DISMM	STRGR	MTVFP	HRKFT	MSTAE	SSQSS	MTPOS	HERKL	
Gütemaße										
Precision (macro avg)	0.52	0.35	0.51	0.78	0.33	0.17	0.45	0.73	0.33	
Precision (weighted avg)	0.84	1.00	0.88	1.00	0.96	0.95	0.99	1.00	0.95	
Recall (macro avg)	0.82	0.99	0.87	0.98	0.69	0.50	0.97	0.78	0.63	
Recall (weighted avg)	0.77	0.97	0.74	0.99	0.86	0.19	0.93	1.00	0.89	
F1_score (macro avg)	0.58	0.42	0.54	0.83	0.37	0.12	0.54	0.72	0.36	
F1_score (weighted avg)	0.78	0.98	0.78	0.99	0.90	0.28	0.95	1.00	0.91	
accuracy	0.77	0.97	0.74	0.99	0.86	0.19	0.93	1.00	0.89	
Zielfelder	SBDKZ	ZZKZ_AL TTEIL	SOBSK	ZZKALKZ	TRAGR	BWPEI	VKORG	DWER K	ZAEHL	LGRAD
Gütemaße										
Precision (macro avg)	0.93	0.63	0.62	0.64	0.50	0.76	0.86	0.74	0.23	0.41

Precision (weighted avg)	0.93	0.87	0.99	0.78	1.00	0.85	1.00	0.98	0.69	1.00
Recall (macro avg)	0.94	0.68	0.71	0.57	0.49	0.79	0.86	0.88	0.30	0.59
Recall (weighted avg)	0.93	0.85	0.92	0.76	0.99	0.72	1.00	0.80	0.54	0.98
F1_score (macro avg)	0.93	0.65	0.50	0.59	0.50	0.72	0.86	0.75	0.22	0.41
F1_score (weighted avg)	0.93	0.86	0.95	0.75	0.99	0.73	1.00	0.87	0.60	0.99
accuracy	0.93	0.85	0.92	0.76	0.99	0.72	1.00	0.80	0.54	0.98

7.3.2 Auswertung der Ergebnisse aus dem CNB – Algorithmus

Nachdem die Features und Targets ausgewählt und die Algorithmen verwendet wurden, ergeben sich in der Tabelle 8 die Auswertung aus dem CNB – Algorithmus. Im Gegensatz zum RFC – Algorithmus gibt mit dem CNB-Algorithmus nur 4 Target (VOLEH, MEABM, BWPEI, ZAEHL) deren mindestwert für die betrachtete Gütemaße unter 80% liegt.

Das Target VOLEH hat eine Gewichtete-Durchschnittspräzision von 75%, ein Gewichtetes – Durchschnittsrecall von 75%, ein Gewichtetes – DurchschnittsF1Score von 75% und eine Genauigkeit von 76%.

Das Target MEABM hat eine Gewichtete-Durchschnittspräzision von 77%, ein Gewichtetes – Durchschnittsrecall von 78%, ein Gewichtetes – DurchschnittsF1Score von 75% und eine Genauigkeit von 78%.

Das Target BWPEI hat ein Gewichtetes – Durchschnittsrecall von 75%, ein Gewichtetes – DurchschnittsF1Score von 76% und eine Genauigkeit von 75%.

Das Target ZAEHL hat eine Gewichtete-Durchschnittspräzision von 71%, ein Gewichtetes – Durchschnittsrecall von 64%, %, ein Gewichtetes – DurchschnittsF1Score von 65% und eine Genauigkeit von 64%.

Die restliche 25 Target (MEINS, KOSGR, VTWEG, SPART, OCMPPF, ZZVERSCHLKZA, EKWSL, GEWEI, DISLS, DISMM, STRGR, MTVFP, HRKFT, MSTAE, SSQSS, MTPOS, HERKL, SBDKZ, ZZKZ_ALTTEIL, SOBSK, ZZKALKZ, TRAGR, VKORG, DWERK, LGRAD) haben für die betrachteten Gütemaße in der Tabelle 8 Werte, die zwischen 80 und 100% liegen.

Tabelle 8: Ergebnisse des CNB des dritten Datensatzes

MTART_1030										
Features Felder	EKGRP	PRCTR	MATKL	WERKS	DISPO	PSTAT				
Features Felder	LABOR	MTART	BKLAS	ZZ_STUEL L_ART	DZEIT	WZEIT				
Algorithmus										
CNB - Algorithmus										
Target	MEINS	KOSGR	VTWEG	VOLEH	SPART	OCMPF	ZZVERSC HLKZA	EKWSL	MEABM	GEWEI
Gütemaße										
Precision (macro avg)	0.33	1.00	0.98	0.44	0.42	0.99	0.64	0.42	0.44	0.67
Precision (weighted avg)	1.00	1.00	0.97	0.75	0.90	1.00	0.86	0.91	0.77	1.00
Recall (macro avg)	0.33	1.00	0.91	0.45	0.34	0.99	0.66	0.42	0.36	0.51
Recall (weighted avg)	1.00	1.00	0.97	0.75	0.92	0.99	0.86	0.91	0.78	1.00
F1_score (macro avg)	0.33	1.00	0.94	0.44	0.36	0.99	0.65	0.43	0.36	0.56
F1_score ((weighted avg)	1.00	1.00	0.97	0.75	0.91	0.99	0.86	0.91	0.75	1.00
accuracy	1.00	1.00	0.97	0.76	0.92	0.99	0.86	0.91	0.78	1.00
Target	DISLS	DISMM	STRGR	MTVFP	HRKFT	MSTAE	SSQSS	MTPOS	HERKL	
Gütemaße										
Precision (macro avg)	0.63	0.33	0.61	0.90	0.41	0.62	0.67	0.66	0.55	
Precision (weighted avg)	0.95	1.00	0.96	1.00	0.95	0.94	0.99	1.00	0.94	
Recall (macro avg)	0.63	0.33	0.62	0.92	0.33	0.39	0.68	0.58	0.52	
Recall (weighted avg)	0.95	1.00	0.96	1.00	0.96	0.96	0.99	1.00	0.94	
F1_score (macro avg)	0.63	0.33	0.61	0.90	0.34	0.42	0.68	0.61	0.52	
F1_score ((weighted avg)	0.95	1.00	0.96	1.00	0.96	0.95	0.99	1.00	0.93	
accuracy	0.95	1.00	0.96	1.00	0.96	0.96	0.99	1.00	0.94	
Target	SBDKZ	ZZKZ_AL TTEIL	SOBSK	ZZKALKZ	TRAGR	BWPEI	VKORG	DWERK	ZAEHL	LGRAD
Gütemaße										
Precision (macro avg)	0.96	0.63	0.49	0.69	0.50	0.77	0.79	0.68	0.26	0.66
Precision (weighted avg)	0.97	0.89	1.00	0.90	1.00	0.85	1.00	0.97	0.71	1.00
Recall (macro avg)	0.97	0.69	0.50	0.64	0.50	0.81	0.79	0.68	0.27	0.67
Recall (weighted avg)	0.97	0.88	1.00	0.90	1.00	0.75	1.00	0.98	0.64	1.00
F1_score (macro avg)	0.97	0.66	0.50	0.66	0.50	0.75	0.79	0.68	0.25	0.66
F1_score ((weighted avg)	0.97	0.66	1.00	0.90	1.00	0.76	1.00	0.97	0.65	1.00
accuracy	0.97	0.66	1.00	0.90	1.00	0.75	1.00	0.98	0.64	1.00

7.4 Auswertung der Ergebnisse des viertens Datensatzes

Für diesen Datensatz wurden 12 Features verwendet und mit diesen Features 22 je als ein Target vorhergesagt. Verwendet wird einen Datensatz mit 30.000 Zeilen und 119 Feldern.

7.4.1 Auswertung der Ergebnisse aus dem RFC – Algorithmus

Nachdem die Features und Targets ausgewählt und die Algorithmen verwendet wurden, ergeben sich in der Tabelle 9 die Auswertungen aus dem RFC – Algorithmus. Diese 5 Targets (ZZARBPL, MMSTA, BWPEI, DZEIT, ZAEHL) haben folgende Gütemaße:

Das Target ZZARBPL hat eine Gewichtete-Durchschnittspräzision von 76% ein Gewichtetes – Durchschnittsrecall von 45%, ein Gewichtetes – DurchschnittsF1Score von 55% und eine Genauigkeit von 45%.

Das Target MMSTA hat ein Gewichtetes – Durchschnittsrecall von 61%, ein Gewichtetes – DurchschnittsF1Score von 72% und eine Genauigkeit von 61%.

Das Target BWPEI hat eine Gewichtete-Durchschnittspräzision von 78% ein Gewichtetes – Durchschnittsrecall von 67%, ein Gewichtetes – DurchschnittsF1Score von 68% und eine Genauigkeit von 67%.

Das Target DZEIT hat ein Gewichtetes – Durchschnittsrecall von 44%, ein Gewichtetes – DurchschnittsF1Score von 53% und eine Genauigkeit von 44%.

Das Target ZAEHL hat eine Gewichtete-Durchschnittspräzision 79% ein Gewichtetes – Durchschnittsrecall von 35%, ein Gewichtetes – DurchschnittsF1Score von 46% und eine Genauigkeit von 35%.

Für diese 5 Felder kann festgestellt werden, dass es mindestens einen Wert für die betrachtete Gütemaße gibt, der unter 80% liegt.

Die restlichen 17 Target (BESKZ , VTWEG, SPART, EKWSL, FHORI, MTVFP, HRKFT, DISMM, ZZ_STUELI_ART, HERKR, LGFSB, HERKL, MTPOS, ZZKZ_ALTTEIL, BASMG, VKORG, LOSGR) haben für die betrachteten Gütemaße in der Tabelle 9 Werte, die zwischen 80 und 100% liegen.

Tabelle 9: Ergebnisse des RFC des vierten Datensatzes

MTART_1040								
Features Felder	EKGRP	PRCTR	MATKL	WERKS	DISPO	PSTAT		
Features Felder	LABOR	MTART	WZEIT	WEBAZ	PLIFZ	DWERK		
Algorithmus								
RFC - Algorithmus								
Target	BESKZ	VTWEG	SPART	EKWSL	FHORI	MTVFP	HRKFT	DISMM
Gütemaße								
Precision (macro avg)	0.94	0.93	0.48	0.51	0.80	1.00	1.00	1.00
Precision (weighted avg)	0.99	0.98	0.87	0.95	0.97	1.00	1.00	1.00
Recall (macro avg)	0.99	0.96	0.69	0.73	0.81	0.99	1.00	1.00
Recall (weighted avg)	0.99	0.98	0.64	0.93	0.93	1.00	1.00	1.00
F1_score (macro avg)	0.96	0.94	0.51	0.54	0.78	1.00	1.00	1.00
F1_score (weighted avg)	0.99	0.98	0.69	0.93	0.95	1.00	1.00	1.00
accuracy	0.99	0.98	0.64	0.93	0.93	1.00	1.00	1.00
Target	ZZ_STUELI_A RT	ZZARBPL	HERKR	LGFSB	HERKL	MTPOS	MMSTA	
Gütemaße								
Precision (macro avg)	0.52	0.31	0.96	0.84	0.79	0.93	0.25	
Precision (weighted avg)	0.89	0.76	0.98	0.97	0.97	0.98	0.96	
Recall (macro avg)	0.55	0.32	0.97	0.95	0.92	0.99	0.56	
Recall (weighted avg)	0.88	0.45	0.98	0.92	0.91	0.98	0.61	
F1_score (macro avg)	0.53	0.26	0.97	0.87	0.79	0.95	0.26	
F1_score weighted avg)	0.88	0.55	0.98	0.94	0.93	0.98	0.72	
accuracy	0.88	0.45	0.98	0.92	0.91	0.98	0.61	
Zielfelder	ZZKZ_ALTTEI L	BWPEI	BASMG	VKORG	DZEIT	LOSGR	ZAEHL	
Gütemaße								
Precision (macro avg)	0.72	0.69	0.50	1.00	0.53	0.38	0.06	
Precision (weighted avg)	0.87	0.76	0.99	1.00	0.85	0.99	0.79	
Recall (macro avg)	0.77	0.71	0.50	1.00	0.65	0.39	0.09	
Recall (weighted avg)	0.85	0.67	0.99	1.00	0.44	0.96	0.35	
F1_score (macro avg)	0.74	0.66	0.50	1.00	0.50	0.39	0.05	
F1_score (weighted avg)	0.86	0.68	0.99	1.00	0.53	0.98	0.46	
accuracy	0.85	0.67	0.99	1.00	0.44	0.96	0.35	

7.4.2 Auswertung der Ergebnisse aus dem CNB – Algorithmus

Nachdem die Features und Targets ausgewählt und die Algorithmen verwendet wurden, ergeben sich in der Tabelle 10 die Auswertungen aus dem CNB–Algorithmus. Diese 3 Targets (ZZARBPL, ZAEHL, BWPEI.), deren mindestwert für die betrachtete Gütemaße unter 80% liegt.

Das Target ZZARBPL hat eine Gewichtete-Durchschnittspräzision 74% ein Gewichtetes – Durchschnittsrecall von 79%, ein Gewichtetes – DurchschnittsF1Score von 78% und eine Genauigkeit von 79%.

Das Target ZAEHL hat eine Gewichtete-Durchschnittspräzision 73% ein Gewichtetes – Durchschnittsrecall von 64%, ein Gewichtetes – DurchschnittsF1Score von 65% und eine Genauigkeit von 64%.

Das Target BWPEI hat eine Gewichtete-Durchschnittspräzision 75%, und ein Gewichtetes – DurchschnittsF1Score von 78%.

Die restliche 19 Target (BESKZ, VTWEG, SPART, EKWSL, FHORI, MTVFP, HRKFT, DISMM, ZZ_STUELI_ART, HERKR, LGFSB, HERKL, MTPOS, MMSTA, ZZKZ_ALTTEIL, BASMG, VKORG, DZEIT, LOSGR) haben für die betrachteten Gütemaße in der Tabelle 10 Werte, die zwischen 80 und 100% liegen.

Tabelle 10: Ergebnisse des CNB des vierten Datennetzes

MTART_1040								
Features Felder	EKGRP	PRCTR	MATKL	WERKS	DISPO	PSTAT		
Features Felder	LABOR	MTART	WZEIT	WEBAZ	PLIFZ	DWERK		
Algorithmus								
CNB - Algorithmus								
Target	BESKZ	VTWEG	SPART	EKWSL	FHORI	MTVFP	HRKFT	DISMM
Gütemaße								
Precision (macro avg)	0.98	0.91	0.79	0.66	0.80	0.99	0.66	1.00
Precision (weighted avg)	0.99	0.98	0.87	0.97	0.99	0.99	1.00	1.00
Recall (macro avg)	0.99	0.98	0.79	0.74	0.79	0.99	0.67	0.97
Recall (weighted avg)	0.99	0.98	0.87	0.98	1.00	0.99	1.00	1.00
F1_score (macro avg)	0.99	0.94	0.77	0.69	0.80	0.99	0.66	0.98
F1_score (weighted avg)	0.99	0.98	0.87	0.97	0.99	0.99	1.00	1.00
accuracy	0.99	0.98	0.87	0.98	1.00	0.99	1.00	1.00
Target	ZZ_STUELI_A RT	ZZARBPL	HERKR	LGFSB	HERKL	MTPOS	MMSTA	
Gütemaße								
Precision (macro avg)	0.54	0.53	0.98	0.96	0.65	0.95	0.37	
Precision (weighted avg)	0.92	0.74	0.98	0.98	0.95	0.99	0.96	
Recall (macro avg)	0.57	0.52	0.98	0.87	0.62	1.00	0.31	
Recall (weighted avg)	0.92	0.79	0.98	0.98	0.96	0.99	0.98	
F1_score (macro avg)	0.55	0.52	0.98	0.89	0.62	0.97	0.33	
F1_score (weighted avg)	0.92	0.76	0.98	0.97	0.95	0.99	0.97	
accuracy	0.92	0.79	0.98	0.98	0.96	0.99	0.98	
Target	ZZKZ_ALTTEIL	BWPEI	BASMG	VKORG	DZEIT	LOSGR	ZAEHL	
Gütemaße								
Precision (macro avg)	0.89	0.67	0.58	1.00	0.53	0.67	0.12	
Precision (weighted avg)	0.91	0.73	1.00	1.00	0.88	1.00	0.75	
Recall (macro avg)	0.72	0.69	0.67	1.00	0.50	0.66	0.08	
Recall (weighted avg)	0.92	0.64	1.00	1.00	0.88	1.00	0.81	
F1_score (macro avg)	0.78	0.64	0.62	1.00	0.51	0.66	0.08	
F1_score (weighted avg)	0.91	0.65	1.00	1.00	0.87	1.00	0.78	
accuracy	0.92	0.64	1.00	1.00	0.88	1.00	0.81	

8 Fazit und Ausblick

In der vorliegenden Arbeit handelt es sich um „Big-Data-Analysen von Materialstammdaten zur Ableitung von Regeln für eine automatisierte Daten-Vorbelegung“. Es ging um die Ermittlung von Feldern, welche eine gegenseitige Abhängigkeit und eine Vorbelegung von Datensätzen ermöglichen. Die Big Data Analyse und der Ansatz des maschinellen Lernens sollten bei der Problemlösung Hilfestellung geben. Im Kapitel 1.3 wurde die Vorgehensweise für die Bearbeitung der Thematik beschrieben.

Beim letzten Teil dieser Arbeit geht es um die Auswertungen der Ergebnisse der vier Datensätze. Die Ergebnisse können folgendermaßen zusammengefasst werden:

Für die vier unterschiedlichen Datensätze können Targets entweder mit dem RFC-Algorithmus oder mit dem CNB - Algorithmus vorhergesagt werden, wobei alle betrachteten Gütemaße für diese Target mindesten 80% betragen müssen. Alle anderen Targets, deren Mindestwert der betrachteten Gütemaße unter 80% liegt, sollten weder mit dem RFC-Algorithmus noch mit dem CNB-Algorithmus vorhergesagt werden. Ein Vergleich der Werte der betrachteten Gütemaße Target pro Target zwischen dem RFC – Algorithmus und dem CNB - Algorithmus zeigt, dass der CNB - Algorithmus bessere Ergebnisse als der RFC - Algorithmus hat. Darüber hinaus können mehr Targets mit dem CNB-Algorithmus als mit dem RFC – Algorithmus vorhergesagt werden. Beispielsweise können für den ersten Datensatz 21 Targets mit dem RFC vorhergesagt werden. Dagegen kann der CNB-Algorithmus mit dem gleichen Datensatz 28 Targets vorhersagen. Für den zweiten Datensatz können 15 Targets mit dem RFC – Algorithmus vorhergesagt werden. Dagegen kann der CNB-Algorithmus mit dem gleichen Datensatz 24 Targets vorhersagen. Für den dritten Datensatz können 18 Targets mit dem RFC – Algorithmus vorhergesagt werden. Dagegen kann der CNB-Algorithmus mit dem gleichen Datensatz 25 Targets vorhersagen. Für den vierten Datensatz können 17 Targets mit dem RFC - Algorithmus vorhergesagt werden. Dagegen kann der CNB-Algorithmus mit dem gleichen Datensatz 19 Targets vorhersagen.

Anhand dieser Ergebnisse kann resümiert werden, dass mit dem CNB-Algorithmus einerseits mehr Targets vorhergesagt werden können und es andererseits im Vergleich zum RFC-Algorithmus bessere Ergebnisse gibt.

Durch diese Arbeit werden folgende Effekte ermöglicht:

- Reduzierung des Arbeitsaufwands bei der Pflege der Materialstammfelder, Denn sobald neue Daten in das System des Konzerns einfließen werden und die 12 Felder befüllt sein werden, werden auch alle anderen Targets direkt gefüllt.

- Verkürzung der Durchlaufzeit bis ein neuer Materialstamm verwendet werden kann, Denn mit diesem Algorithmus werden die Materialstämme den Mitarbeitern in den verschiedenen Abteilungen schneller zur Verfügung stehen. Darüber hinaus werden die Fehlerquellen durch manuelle Eingabe reduziert, Denn die Befüllung von Feldern wird automatisch durch den Algorithmus erfolgen.

In dieser Arbeit wurden zwei Algorithmen zum Erreichen des Ziels verwendet und ihre Ergebnisse wurden ausgewertet. Diese zwei Algorithmen wurden ausgewählt, weil sie mit unausgewogenen Klassen in den Daten gut umgehen können.

Generell gibt es mehrere Methoden, die zur Lösung des Problems der Unausgewogenheit auf die Daten angewendet werden könnten. Beispielsweise könnten mehr Daten gesammelt werden: Ein größerer Datensatz könnte eine andere und vielleicht ausgewogenere Perspektive auf die Klassen aufzeigen, was die Ergebnisse der verwandten Algorithmen verbessern könnte. Außerdem könnte die Frage gestellt werden, ob eine Erhöhung der Anzahl der Features die erzielten Ergebnisse noch verbessern könnte.

9 Literaturverzeichnis

- Gillhuber, A. (2019). *Automatisierte Datenanalyse in Echtzeit*.
<https://www.industrial-production.de/ki---datenanalyse/das-maximum-erreichen.htm>
- Grandini, M., Bagli, E. & Visani, G. (2020). *METRICS FOR MULTI-CLASS CLASSIFICATION: AN OVERVIEW*. <https://arxiv.org/pdf/2008.05756.pdf>
- Hatwell, J., Medhat Gaber, M. & Atif Azad, R. M. (2020). Vol.:(0123456789) *Artificial Intelligence Review (2020) 53:5747–5788* Vol.:(0123456789) *Artificial Intelligence Review (2020) 53:5747–5788* CHIRPS: Explaining random forest classification.
<https://link.springer.com/content/pdf/10.1007/s10462-020-09833-6.pdf>
- ICHI.PRO. (2020 - 2021). *Datenvorverarbeitung*.
<https://ichi.pro/de/datenvorverarbeitung-237599058119091>
- MDIS. (2020). *Big Data zur Analyse & Optimierung im Unternehmen*.
<https://www.mdis-consulting.de/big-data.html>
- O'REILLY. (2021). *The elbow method*.
<https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml>
- Paluv, R. (2018). *Grundlagen des Machine Learning – überwachtes und unüberwachtes Lernen*. <https://plus-it.de/blog/machine-learning-ueberwachtes-vs-unueberwachtes-lernen/>
- Rennie, J. D. M., Shih, L., Teevan, J. & Karger, D. R. (2003). *Tackling the Poor Assumptions of Naive Bayes Text Classifiers*.
<https://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf>
- Saint-Cirgue, G. (2019). *Apprendre le Machine learning en une semaine*.
https://gallery.mailchimp.com/3388ea9e390699643fbc661f1/files/57c9101e-5e64-4f43-b0fd-64aeaa9f9bc/Apprendre_le_ML_en_une_semaine.pdf
- SAP. (o.J.). *Maschinelles Lernen mit künstlicher Intelligenz*.
<https://www.sap.com/germany/insights/what-is-machine-learning.html>

Scikit-learn developers. (2007 - 2020a). *Metrics and scoring: quantifying the quality of predictions*.
https://scikitlearn.org/stable/modules/model_evaluation.html#classification-metrics

Scikit-learn developers. (2007 - 2020b). *Pipelines and composite estimators*.
<https://scikit-learn.org/stable/modules/compose.html>

Scikit-learn developers. (2007 - 2020c). *sklearn.naive_bayes.ComplementNB*.
https://scikitlearn.org/stable/modules/generated/sklearn.naive_bayes.ComplementNB.html

Scikit-learn developers. (2007 - 2020d). *sklearn.preprocessing.OneHotEncoder*.
<https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>

V.S.Thiyagarajan & K.Venkatachalapathy. (2014). *ISOLATING VALUES FROM BIG DATA WITH THE HELP OF FOUR V'S*.
<https://ijret.org/volumes/2015v04/i01/IJRET20150401022.pdf>

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data* (First edition). O'Reilly.

Anhang

a) Anhang

Tabelle 11: Bezeichnungen von den verwendeten Feldern (Features, Target)

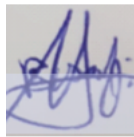
Felder	Bezeichnungen
MEINS	Basismengeneinheit
BESKZ	Beschaffungsart
OCMPF	Gesamtprofil für Auftragsänderungsdienst
KOSGR	Gemeinkostengruppe der Kalkulation
VTWEG	Vertriebsweg
KLART	Klassenart
VOLEH	Volumeneinheit
MSTAE	Werksübergreifender Materialstatus
MAGRV	Materialgruppe Packmittel
SSQSS	Steuerschlüssel für Qualitätsmanagement in der Beschaffung
HERKL	Ursprungsland des Materials (IHK-Ursprung)
MMSTA	Werksspezifischer Materialstatus
AWSLS	Abweichungsschlüssel
MTPOS	allgemeine Positionstypengruppe
HERKL	Ursprungsland des Materials (IHK-Ursprung)
ZZ_STUELI_ART	Krones Feld
ZZARBPL	Krones Feld
HERKR	Ursprungsregion des Materials (IHK-Ursprung)
LGFSB	Vorschlagslagerort für Fremdbeschaffung
HERKL	Ursprungsland des Materials (IHK-Ursprung)
SPART	Sparte
ZZVERSCHKZA	Krones Feld
EKWSL	Einkaufswerteschlüssel
MEABM	Einheit für Länge/Breite/Höhe
GEWEI	Gewichtseinheit
TRAGR	Transportgruppe
DISLS	Dispositionslosgröße
DISMM	Dispositionsmerkmal
FHORI	Horizontschlüssel für Pufferzeiten
STRGR	Planungsstrategiegruppe
MTVFP	Prüfgruppe für Verfügbarkeitsprüfung
VINT1	Verrechnungsintervall -Rückwärts
VINT2	Verrechnungsintervall -Vorwärts-
HRKFT	Herkunftsguppe als Untergliederung der Kostenart
ZZAUSLOESER	Krones Feld
MFRNR	Nummer eines Herstellers
MATKL	Warengruppe
WERKS	Werk
DISPO	Disponent

Felder	Bezeichnungen
PSTAT	Pflegestatus
ZZTEXTN	Krones Feld
ZZGEWUSER	Krones Feld
LABOR	Labor/Konstruktionsbüro
EKGRP	Einkäufergruppe
PRCTR	Profitcenter
MEINS	Basismengeneinheit
BESKZ	Beschaffungsart
OCMPF	Gesamtprofil für Auftragsänderungsdienst
KOSGR	Gemeinkostengruppe der Kalkulation
VTWEG	Vertriebsweg
KLART	Klassenart
VOLEH	Volumeneinheit
MSTAE	Werkübergreifender Materialstatus
MAGRV	Materialgruppe Packmittel
SSQSS	Steuerschlüssel für Qualitätsmanagement in der Beschaffung
HERKL	Ursprungsland des Materials (IHK-Ursprung)
MMSTA	Werksspezifischer Materialstatus
AWSLS	Abweichungsschlüssel
MTPOS	allgemeine Positionstypengruppe
HERKL	Ursprungsland des Materials (IHK-Ursprung)
ZZ_STUEL_ART	Krones Feld
ZZARBPL	Krones Feld
HERKR	Ursprungsregion des Materials (IHK-Ursprung)
LGFSB	Vorschlagslagerort für Fremdbeschaffung
HERKL	Ursprungsland des Materials (IHK-Ursprung)
SPART	Sparte
ZZVERSCHKZA	Krones Feld
EKWSL	Einkaufswerteschlüssel
MEABM	Einheit für Länge/Breite/Höhe
GEWEI	Gewichtseinheit
TRAGR	Transportgruppe
DISLS	Dispositionslosgröße
DISMM	Dispositionsmerkmal
FHORI	Horizontschlüssel für Pufferzeiten
STRGR	Planungsstrategiegruppe
MTVFP	Prüfgruppe für Verfügbarkeitsprüfung
VINT1	Verrechnungsintervall -Rückwärts
VINT2	Verrechnungsintervall -Vorwärts-
HRKFT	Herkunftsguppe als Untergliederung der Kostenart
ZZAUSLOESER	Krones Feld
MFRNR	Nummer eines Herstellers
MATKL	Warengruppe
WERKS	Werk
DISPO	Disponent
PSTAT	Pflegestatus
ZZTEXTN	Krones Feld
ZZGEWUSER	Krones Feld

Felder	Bezeichnungen
LABOR	Labor/Konstruktionsbüro
EKGRP	Einkäufergruppe
PRCTR	Profitcenter

Erklärung

1. Mir ist bekannt, dass dieses Exemplar der Bachelorarbeit/Masterarbeit als Prüfungsleistung in das Eigentum der Ostbayerischen Technischen Hochschule Regensburg übergeht.
2. Ich erkläre hiermit, dass ich diese Bachelorarbeit/Masterarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.



Regensburg, den 28.08.2021

Ort, Datum und Unterschrift