Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Tobias Braack

Miningverfahren zur Clusterbildung technischer Artefakte

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Tobias Braack

Miningverfahren zur Clusterbildung technischer Artefakte

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Dr.-Ing. Sabine Schumann

Eingereicht am: 4. April 2017

Tobias Braack

Thema der Arbeit
Miningverfahren zur Clusterbildung technischer Artefakte

Stichworte
Data-Mining, Cluster-Analyse, Knowledge Discovery in Databases, Technische Artefakte, Datenvorverarbeitung

Kurzzusammenfassung
In dieser Arbeit wird eine Cluster-Analyse technischer Artefakte anhand des Knowledge-Discovery-in-Databases-Prozesses durchgeführt. Nach der Vorstellung dieses Prozesses, wird dieser am Beispiel des Gegenstandbereiches der Schrauben praktisch durchlaufen. Es werden die Probleme der einzelnen Phasen thematisiert, sowie geeignete Lösungen für diese präsentiert. Nach Durchführung der Cluster-Analyse, wird eine Bewertung der Resultate durch einen Fachexperten vorgenommen. Zudem werden die verwendeten Verfahren der Attribut-Selektion, sowie der Cluster-Analyse gegenübergestellt und bewertet. Ebenso wird eine Bewertung hinsichtlich der Übertragbarkeit auf andere Bereiche vorgenommen.
Tobias Braack

Title of the paper
Mining procedure for clustering technical artefacts

Keywords
Data-Mining, Cluster analysis, Knowledge Discovery in Databases, Technical artefacts, Data preprocessing

Abstract
In this thesis, a cluster analysis of technical artefacts is performed using the Knowledge Discovery in Databases process. According to describing the concept of this process, it is practically preformed by the example of the object of screws. The issues and proper solutions for this, will be presented for every phase. After running cluster analysis, the results will be valuated by an expert. Furthermore the used procedures for attribute selection and data analysis will be discussed and valuated. The portability to other objects and sections will also be reviewed.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Listings

1. Einleitung

1.1. Motivation

Bereits in den 1940er und 1950er Jahren wurden die Grundsteine der *künstlichen Intelligenz* (engl. artificial intelligence, Abk. AI) gelegt. Doch erst seit einigen Jahren erlebt die *KI* in Form des *maschinellen Lernens* eine Renaissance. Mit zunehmend günstigerer Hardware, sowie deren Verfügbarkeit in der *Cloud* lassen sich nun KI-Anwendungen realisieren, die vor einigen Jahren noch mangels Rechenleistung gescheitert sind. Zudem hat die Verfügbarkeit von Daten in großen Mengen, etwa durch Sensoren, Smartphones oder dem Internetverhalten, wie der Nutzung von sozialen Netzwerken oder Online-Shopping-Plattformen, einen Anteil an dem Aufschwung des maschinellen Lernens. Das Potential des maschinellen Lernens blieb natürlich nicht unentdeckt und große Konzerne wie Amazon, Facebook, Google, IBM und Microsoft begannen in Forschung und Entwicklung zu investieren. Das maschinelle Lernen unterscheidet sich von den herkömmlichen Methoden durch Algorithmen, die nicht wie bisher speziell auf einen Anwendungszweck kodiert werden. Vielmehr werden Algorithmen eingesetzt, die beispielsweise anhand von Trainingsdaten hinzulernen. Das Ziel ist es, durch die Trainingsdaten bekannte Muster zu erkennen und Zusammenhänge herzustellen. Daraus entsteht eine Funktion bzw. verändert sich eine Funktion während der Lernphase so, dass die erkannten Muster und Zusammenhänge auch in neuen Daten erkannt werden können. Im Allgemeinen wird durch das maschinelle Lernen analog zum Menschen Wissen aus Erfahrungen erzeugt und daraus Lösungen erstellt. Viele der mittlerweile alltäglichen Anwendungen basieren bereits auf den Methoden des maschinellen Lernens, wie etwa die Spracherkennungssysteme von Apple oder Microsoft, das Suchmaschinen-Ranking von Google, die Gesichtserkennung von Facebook oder die automatische Spam Erkennung bei E-Mail Anwendungen. Über Business-Daten können personalisierte Werbungen erstellt oder abwanderungswillige Kunden identifiziert werden, woraus eine höhere Kundenbindung und Kundenzufriedenheit entstehen kann. Das Anwendungsgebiet des maschinellen Lernens ist also breit gefächert und findet nahezu in jeder Branche Verwendung.
Eng verwandt mit dem Bereich des maschinellen Lernens ist das *Data-Mining* bzw. das *Know-*

ledge Discovery in Databases (KDD), die immer wieder als Synonyme verwendet werden. Dabei handelt es sich beim *KDD* um einen Prozess, der neben dem Schritt des Data-Minings auch die Schritte der Datenvorverarbeitung und der Bewertung der Resultate umfasst. Das Ziel ist das Finden von bislang unbekannten Mustern und Zusammenhängen in großen Datenbeständen. Der Übergang vom maschinellen Lernen zum Data-Mining ist recht unscharf, da die Klassifizierung oftmals als Data-Mining-Aufgabe angesehen wird, jedoch auch den Zielen des maschinellen Lernens entspricht. Viele der eingesetzten Verfahren im Bereich des Data-Minings entstammen der Statistik, doch auch Verfahren des maschinellen Lernens finden Anwendung, wodurch wohl auch keine klare Trennung dieser Bereiche möglich ist. Durch die Verwendung von statistischen Verfahren, die an die Komplexität des Data-Minings angepasst sind, entstehen nur unscharfe und ungefähre Ergebnisse, die jedoch der Anwendung im Data-Mining häufig genügen. Als eine typische Aufgabe des Data-Minings wird immer wieder die Identifizierung der Kreditwürdigkeit von Kunden erwähnt. Dazu wird ein Klassifikator erstellt, der auf Basis von bereits als *kreditwürdig* und *kreditunwürdig* klassifizierten Kundendaten trainiert wird. Über diesen Klassifikator können zukünftig Kunden hinsichtlich ihrer Kreditwürdigkeit bewertet werden. Neben der Klassifizierung umfasst das Data-Mining unter anderem auch Abhängigkeitsanalysen. Dabei kann beispielsweise durch Warenkorb-Analysen ermittelt werden, dass Kunden, die eine Flasche Schnaps kaufen, sehr häufig auch Softgetränke kaufen, aber der Kauf von Softgetränken nicht unbedingt den Kauf von Schnaps nach sich zieht. Unter Berücksichtigung dieses Wissens könnte eine (Re-) Platzierung der Produkte vorgenommen werden. Eine weitere Disziplin des Data-Minings ist die Cluster-Analyse. Bei der Cluster-Analyse geht es darum, eine Menge von Objekten in homogene Gruppen bzw. Cluster zu zerlegen. Dabei sollen sich Objekte unterschiedlicher Cluster möglichst unähnlich und gleicher Cluster möglichst ähnlich sein. Im Grunde genommen handelt es sich dabei um eine Klassifizierung, die allerdings ohne das Training mit bereits klassifizierten Daten durchgeführt wird. Eine Anwendung der Cluster-Analyse könnte die der Kundensegmentierung sein, um individuell auf die Interessen der einzelnen Gruppen eingehen und werben zu können. Eine weitere Anwendung könnte in der Gruppierung von Produkten liegen.
In dem Unternehmen *Claudius Peters Projects GmbH*, einem Spezialisten für Schüttgut- und Verfahrenstechnik in der Zement-, Kohle-, Aluminium-, Gips- und Schüttgutindustrie, hat sich der Datenbestand der Teile im Laufe der Jahre um eine Vielzahl von Klassen und Attributen angereichert. Aufgrund der geplanten Einführung eines neuen Systems, sowie der allgemeinen Hinterfragung nach der Sinnhaftigkeit und Aktualität dieser Gruppierungen, entstand die Anforderung nach einer, unabhängig von der bestehenden Klassifizierung, intelligenten Methode zur Gruppierung der Teile. Eine manuelle Sichtung der Daten durch einen

Mitarbeiter kommt aufgrund der Datenmenge nicht infrage, zudem unter Umständen keine Veränderungsmöglichkeiten wegen Betriebsblindheit entdeckt werden würden. Daher wurde sich für die Durchführung einer Cluster-Analyse technischer Artefakte entschieden, wodurch die Entstehung neuartiger Gruppierungen, auf Basis entdeckter, bisher unbekannter Muster und Zusammenhänge in den Daten, möglich ist. Neben den Anforderungen des Unternehmens liegt die Motivation dieser Arbeit in einer persönlichen Neugier für das Informatikspezialgebiet der künstlichen Intelligenz, insbesondere für die Themenbereiche des Data-Minings, sowie des maschinellen Lernens.

1.2. Ziele

Das Ziel dieser Arbeit ist, anhand der praktischen Anwendung des KDD-Prozesses zu überprüfen, ob dieser in Bezug auf die Durchführung einer Cluster-Analyse technischer Artefakte geeignet ist. Dafür ist zunächst die Überprüfung der Strukturierung der Daten hinsichtlich deren Verwendbarkeit für das Data-Mining-Verfahren nötig. Sind diese für die Verwendung nicht geeignet, gilt es die Qualität der Daten durch angemessene Maßnahmen zu steigern und diese in die für das Data-Mining-Verfahren benötigten Form zu bringen. Mittels den aus der Cluster-Analyse erzielten Ergebnissen sind die Möglichkeiten der Gruppierungen darzustellen und somit eine Sensibilisierung der Fachgebietsexperten vorzunehmen. Ist die Sensibilisierung nicht erfolgreich und die Bewertung des Resultats fällt negativ aus, sind die Gründe zu analysieren und entsprechende Anpassungen oder weitere Maßnahmen an den Daten und eine erneute Cluster-Analyse durchzuführen.

1.3. Aufbau der Arbeit

Der Aufbau dieser Arbeit untergliedert sich in fünf Kapitel. Nach einer kurzen Einführung wird im Kapitel 2 *Analyse* der Aufbau der zugrundeliegenden Daten, sowie deren Charakteristika beschrieben (2.1). Ebenso umfasst dieses Kapitel die Grundlagen des in dieser Arbeit verwendeten KDD-Prozesses (2.2). Dabei werden die einzelnen Phasen des Prozesses anhand möglicher Problemstellungen und Lösungen näher gebracht. Zum Abschluss dieses Kapitels folgt eine konkretisierte Fassung der Aufgabenstellung (2.3).
Das Kapitel 3 umfasst die *praktische Durchführung* des KDD-Prozesses. Zunächst wird eine Auswahl an möglichen Programmen zur Umsetzung der Aufgabe, sowie die letztendliche Wahl vorgestellt (3.1). Das Kapitel untergliedert sich im Weiteren in die einzelnen Phasen des KDD-Prozesses, die durch die in den Daten aufgetretenen Probleme, sowie deren Reaktionen darauf

beschrieben werden. Das Kapitel 4 *Evaluation* enthält die Bewertung der erzielten Ergebnisse der Cluster-Analysen. Das Kapitel ist nach Durchläufen unterteilt und enthält neben den vorgenommenen Maßnahmen zur Erzielung eines verwertbaren Resultats auch Beispiele, die die Bewertung der Resultate des Durchlaufes durch den Fachgebietsexperten nachvollziehbar machen sollen. Zum Abschluss dieses Kapitels folgt eine Gegenüberstellung der verwendeten Verfahren (4.4).
Im Kapitel 5 *Fazit und Ausblick* schließt die Arbeit mit einer Bewertung hinsichtlich der Ziele und einem Blick über die Anwendung dieser Arbeit hinaus ab.

2. Analyse

Das Kapitel der Analyse beginnt zunächst mit der Darstellung und Beschreibung der Strukturen der vorliegenden Datenbasis, sowie deren Charakteristika. Im Anschluss daran folgen die Grundlagen des KDD-Prozesses. Es werden die typischen Probleme der einzelnen Phasen beschrieben, sowie Möglichkeiten zur Lösungen dieser präsentiert. Eine detaillierte Beschreibung der Aufgabe bringt dieses Kapitel zum Abschluss.

2.1. Datenbasis

Die dieser Arbeit zugrunde liegende Datenbasis besteht aus 7.515 strukturierten Datentabellen mit zum Großteil unterschiedlichen Tabellendefinitionen. Die Datentabellen sind in Klassen strukturiert und repräsentieren eine spezielle Art von Teilen. Jede dieser Klassen besitzt gewisse Attribute, die sich von Klasse zu Klasse sowohl in der Anzahl als auch in der Auswahl der Attribute unterscheiden. Liegt beispielsweise eine Klasse der *Getriebemotoren* vor, hat diese andere Attribute als eine Klasse der *Sechskantschrauben*, da dies grundlegend unterschiedliche Teile sind. Bei Betrachtung der Attribute existieren klassenübergreifend 7.125 einzigartige Attribute. Zudem liegen insgesamt 460.687 Datensätze von Teilen jeglicher Art vor.
Zur Veranschaulichung der Daten werden einige Klassen mit ihren Attributen und einem Auszug von Datensätzen dargestellt. Begonnen wird mit einer Klasse von Getriebemotoren, die in Tabelle 2.1 dargestellt ist.

Teile-Nr	128278-01	128278-02	128278-03
Teilebezeichnung	Stirnrad-Getriebemotor	Stirnrad-Getriebemotor	Stirnrad-Getriebemotor
TYGEMO alpha	R77R37MT90S8-BMG	R77R37MT90S8-BMG	R77R37MT90S8-BMG
ABTDRZ numer.	2,600	2,6	2,7
UEBVER alpha	289	289	276
BAUFRM alpha	V6	V6	V6
ABTLAG alpha	—	—	—
LAGKLK alpha	0	0	0
LAGKAB alpha	normal	normal	normal
DREHMO numer.	386	386	400
WELZDM numer.	40	40	40
LAWEZA numer.	80	80	80
SCHLSG numer.	0,298	0,298	0,35
BTRSPG alpha	400	380	230/400
TOLBSP alpha	± 10	—	± 10
FREQUZ alpha	50	50	50
SCHAAR alpha	Stern	Stern	Dreieck/Stern
NENNST numer.	0,43	0,46	0,43
ISOKLS alpha	F	F	F
UMGTEM numer.	-25 - 40	-25 - 40	-25 - 40
MIUMTE numer.	-25,00	-25	-25
SCHART numer.	65	65	65
BRMSPG numer.	400	220	230
MOMBRS numer.	10	10	10
HOLAGE numer.	0 - 1000	0 - 1000	0 - 1000
EXSART alpha	nein	nein	nein
GETART alpha	Stirnrad	Stirnrad	Stirnrad
DICHWE alpha	Stand.-Mat	Stand.-Mat	Stand.-Mat
HERGEM alpha	SEW	SEW	SEW

Tabelle 2.1.: Klasse 000418: Getriebemotor..MTm.Welle,.mBremse

Wie der Tabelle zu entnehmen ist, bestehen die Attributnamen aus einem Kürzel gefolgt von einem Datentypen. Aus den Metadaten kann eine Beschreibung der abgekürzten Attribut-

namen entnommen werden. Das Attribut *TYGEMO* steht beispielsweise für *Typ Getriebemotor*. Der Datentyp eines Attributs ist entweder mit *alpha* oder *numer.* angegeben. Bei *alpha* handelt es sich um alphanumerische Attributwerte. Trägt ein Attributname *numer.* hinter dem Attributkürzel, sollen die Attributwerte aus numerischen Werten bestehen. Verlässlich sind die Angaben in den Attributnamen aber nicht, denn beispielsweise handelt es sich bei dem Attribut *UMGTEM numer.* genau genommen um alphanumerische Werte, da die numerischen Werte durch ein Bindestrich getrennt als Wertebereich angegeben sind. Wie eingangs bereits erwähnt, haben unterschiedliche Klassen unterschiedliche Attribute. Dadurch besteht eine Heterogenität der Datentabellen. Um dies ersichtlich zu machen, ist eine Datentabelle des Gegenstandsbereich der Schrauben in Tabelle 2.2 dargestellt.

Teile-Nr	011638-01	011644-01	011670-03
Teilebezeichnung	Sechskantschraube M 5 x 20	Sechskantschraube M 6 x 20	Sechskantschraube M10 x 35
SICHNA alpha	—	—	—
GEWDUR alpha	M_5	M_6	M10
LANGE numer.	20	20	35
SCHLUW numer.	8	10	16
KOPFHH numer.	3,5	4	6,4
FESTWE alpha	4.6	4.6	4.6
OBFBEH alpha	gal.verz.	gal.verz.	feu.verz.
ABVOST alpha	gestempelt	gestempelt	gestempelt

Tabelle 2.2.: Klasse 000528: Skt-Schraube Gew.b.K.(C) ISO 4018

Diese Tabelle enthält die Klasse der Sechskantschrauben nach der Norm ISO 4018. Die Tabelle der Sechskantschrauben unterscheidet sich von der Tabelle 2.1 der Getriebemotoren in fast jedem Attribut. Eine gemeinsame Basis bilden dabei nur die Attribute der *Teile-Nr* und der *Teilebezeichnung*. Da bei Motoren andere Eigenschaften zur Beschreibung eines Teils als bei einer Schraube benötigt werden, erscheinen diese unterschiedlichen Attribute der Datentabellen durchaus sinnvoll. Jedoch unterscheiden sich die Tabellendefinitionen auch innerhalb eines Gegenstandsbereiches, wie beispielsweise der Sechskantschraube. Bei Betrachtung der Tabelle 2.2 und der Tabelle 2.3 fällt auf, dass beispielsweise das Attribut *SICHNA alpha* bei letztgenannter Tabelle nicht vorhanden ist.

Teile-Nr	011956-01	011957-01	011958-01
Teilebezeichnung	Sechskantschraube M12 x 50	Sechskantschraube M16 x 50 -SW18-	Sechskantschraube M16 x 60
GEWDUR alpha	M12	M16	M16
NENNLA numer.	50	50	60
SCHLUW numer.	17	18	19
FORM alpha	Ri	Ri	Ri
KOPFHH numer.	9	11	11
FESTWE alpha	14H	22H	22H
OBFBEH alpha	gal.verz.	gal.verz.	gal.verz.
ABVOST alpha	gestempelt	gestempelt	gestempelt

Tabelle 2.3.: Klasse 000529: Skt-Schraube m.Zapfen DIN 561

Es handelt sich also nicht nur bei unterschiedlichen Arten von Teilen, sondern auch bei gleichartigen, um heterogene Datentabellen. Dabei spiegelt sich die Heterogenität nicht nur in den unterschiedlichen Tabellendefinitionen wieder. Die Datentabellen weisen auch synonyme Attribute, das heißt Attribute mit unterschiedlichen Namen aber gleicher Bedeutung, auf. Nachdem der Aufbau der unterschiedlichen Datentabellen beschrieben wurde, sollen nun die Attribute mit ihren Werten charakterisiert werden. Da wären die numerischen Attribute, deren Werte grundsätzlich ohne Einheiten angegeben werden. Bei einigen Attributen ist die Einheit in den Metadaten des Attributs angegeben, wie etwa bei dem Merkmal *LANGE numer.* aus Tabelle 2.2 (siehe Anhang A.1). Bei anderen Merkmalen wiederum ergeben sich die Einheiten aus den handelsüblichen Angaben. Das Dezimaltrennzeichen ist ein Komma und Werte wie *4.6* bei dem Attribut *FESTWE alpha* der Tabelle 2.2, die eine andere Notation von Dezimalzahlen darstellen könnten, sind nicht als solche zu interpretieren. Neben den numerischen Attributen gibt es wie bereits erwähnt auch alphanumerische. Die alphanumerischen Attribute sind, ausgenommen von technischen Angaben wie der des Gewindes einer Schraube *GEWDUR alpha* in Tabelle 2.3, größtenteils in deutscher Sprache, aber auch zum Teil in englischer Sprache verfasst. Zudem weisen diverse Attributwerte Abkürzungen, wie im Attribut *OBFBEH alpha* in Tabelle 2.3, auf. Ebenfalls existieren in den Datentabellen nicht gepflegte bzw. leere Werte. Bei einigen Attributen, über die zum Zeitpunkt der Datenanlage keine Informationen vorlagen, wurde auch, anstatt keinem Wert, ein Wert bestehend aus einem oder mehrerer Minus-Zeichen (-, –, —) angegeben.

2.2. Schritte des Knowledge Discovery in Databases

Der KDD-Prozess beschreibt ein Vorgehen, dass das Ziel der Wissensgewinnung aus Datenbeständen bis hin zur Entdeckung neuer Zusammenhänge hat. Dieser Prozess wurde durch Fayyad geprägt und wie folgt definiert:

> Knowledge Discovery in Databases describes the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.
>
> Fayyad u. a. (1996)

Der KDD ist demnach ein Prozess zur (semi-) automatischen Extraktion von verständlichem, neuen, sowie potentiell nützlichem Wissen aus Datenquellen. Ebenso handelt es sich um einen interaktiven, sowie iterativen Prozess, da vom Anwender Entscheidungen zu treffen sind und einige der Phasen dieses Prozesses mehrmals durchlaufen werden. Neben dem KDD-Prozess existieren auch alternative Modelle zur Durchführung von Datenanalysen, wie zum Beispiel das CRISP[1]-Data-Mining-Modell, auf die hier nicht näher eingegangen wird. Fayyad unterteilt den KDD-Prozess in seiner Arbeit Fayyad u. a. (1996) in 5 Phasen (siehe Abbildung 2.1). In der ersten Phase der *Datenselektion* werden die Rohdaten aus den Datenquellen ausgewählt. Die dann vorliegenden Zieldaten werden in der Phase der Vorverarbeitung bereinigt. In der dritten Phase werden die Daten schließlich transformiert, ehe sie in der vierten Phase mittels eines für den Anwendungszweck geeigneten Data-Mining-Verfahrens analysiert werden. In der fünften Phase werden die dann vorliegenden Ergebnisse hinsichtlich der Ziele evaluiert.

[1] Cross Industry Standard Process

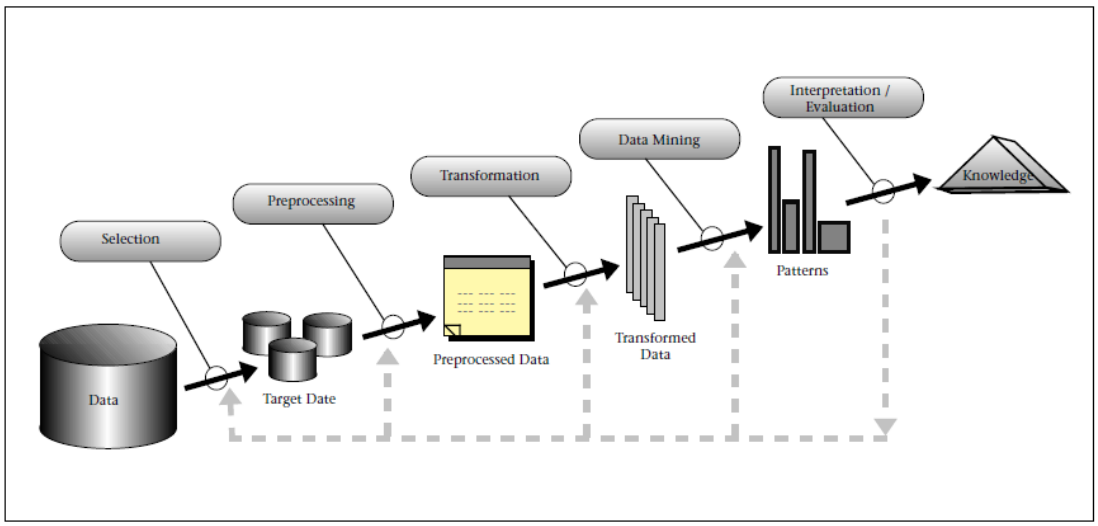


Abbildung 2.1.: KDD Prozess nach Fayyad. Abbildung S. 41 Fayyad u. a. (1996).

Oftmals werden die Begriffe *Data-Mining* und *Knowlegde Discovery in Databases* als Synonym verwendet, jedoch ist das Data-Mining genau genommen ein Teilschritt des KDD, wie der Abbildung 2.1 zu entnehmen ist.
Im Folgenden dieses Abschnittes sollen die 5 Phasen detaillierter beschrieben und auf typische Probleme, sowie Möglichkeiten zur Lösung dieser eingegangen werden. An dieser Stelle sei erwähnt, dass es sich dabei nur um einen Teil der möglichen Probleme und Techniken zur Lösung handelt. Orientiert wurde sich an der Literatur Cleve und Lämmel (2016), die für einen tieferen Einstieg an dieser Stelle empfohlen wird.

2.2.1. Datenselektion und -integration

Die Datenselektion ist die erste Phase des KDD-Prozesses. In dieser Phase geht es darum, eine Auswahl der zu analysierenden Daten aus einem Datenbestand zu treffen. Damit das gelingt, ist es zunächst nötig ein Verständnis für die Anwendung, sowie dem vorhandenen Anwendungswissen zu erlangen. Auf Basis dessen wird das Ziel des KDD aus Anwendungssicht definiert. Zur Phase der Datenselektion gehört auch die Datenintegration. Die Datenintegration beschreibt das Zusammenfügen von Daten aus mehreren Quellen in einen für Analysezwecke zentralen Datenbestand. Dabei können einige Probleme, die exemplarisch an der Tabelle 2.4 erläutert werden sollen, auftreten. Es ist beispielsweise möglich, dass das sogenannte Entitätenidentifikationsproblem auftreten kann. Dabei handelt es sich um Attribute mit unterschiedlichen Attributnamen, die sich jedoch in der Semantik des Attributs gleichen. Die Attribute *Strasse* aus Tabelle A und *Kunden_Str* aus Tabelle B liefern ein Beispiel für ein Entitätenidentifika-

tionsproblem. Diese Attribute beschreiben die gleiche Sache, und zwar eine Straße, wurden jedoch namentlich unterschiedlichen Attributen zugeordnet. Es gilt also diese Attribute zu identifizieren und zusammenzuführen. Semantisch äquivalente Attribute können beispielsweise durch die Metadaten identifiziert werden. Weitere Konflikte können in den Datenwerten eines Attributes auftreten. Im Falle des Attributs *Kontakt* ist in Tabelle A eine E-Mail Adresse hinterlegt, wohingegen dieses Attribut in Tabelle B eine Telefonnummer enthält. Hier gilt es die Werte gemäß ihrer Bedeutung zu trennen. Ebenfalls können Datenwertkonflikte durch unterschiedliche Maßeinheiten, wie beispielsweise die Angabe einer Zeitmessung in Sekunden und Minuten, entstehen. Diese sollten auf eine gemeinsame Einheit überführt werden.

Tabelle 2.4.: Probleme bei der Datenintegration

Kunde	**Strasse**	**Kontakt**
Meier	Turmstraße 1	meier1@mail.de
Schulz	Poststraße 12	schulz@mail.de

Tabelle A: Kundendaten

Kunde	**Kunden_Str**	**Kontakt**
Mueller	Schlossallee 123	015112312121
Reus	Hauptstraße 11	017163179601

Tabelle B: Kundendaten

Beim Zusammenführen von Daten verschiedener Quellen kann es vorkommen, dass diese widersprüchlich sind. Das heißt vermeintlich gleiche Datensätze besitzen für ein Attribut unterschiedliche Inhalte. Beispielsweise sind für denselben Kunden unterschiedliche Wohnorte hinterlegt. Zur Beseitigung von Widersprüchen ist der korrekte oder wahrscheinlichste Wert zu ermitteln und zu übernehmen.
Die Ursache der beschriebenen Probleme besteht in der Zusammenführung von Daten aus unterschiedlichen Datenquellen. Zudem sei erwähnt, dass in diesem Abschnitt nur einige der möglichen Probleme behandelt wurden. Nach Abschluss der Datenselektion und -integration liegt idealerweise eine konsistente Datentabelle mit plausiblen Datensätzen vor.

2.2.2. Datenvorverarbeitung und -bereinigung

Die Zieldaten liegen nach der Selektion bzw. Integration oftmals nicht in der für die Datenanalyse benötigten Qualität vor. In der zweiten Phase des KDD-Prozesses ist also zunächst die Qualität des Datenbestandes zu prüfen, um diese gegebenenfalls durch geeignete Maßnahmen zu steigern. Dieser Schritt ist für ein möglichst unverfälschtes und zuverlässiges Resultat nötig. Das Ziel der Datenvorverarbeitung und -bereinigung besteht also darin eine einheitliche Struktur und Form der Daten zu erlangen oder im Allgemeinen die Datenqualität zu steigern.

Um dies zu erreichen, können die Daten unter Anderem auf die folgenden typischen Anzeichen für fehlerhafte Daten überprüft werden:

- Fehlende Daten
- Falsche Daten
- Syntaktische Probleme
- Semantische Probleme

Fehlende Daten sind einfach zu identifizieren. Dabei sollte zunächst ermittelt werden, ob das Fehlen eines Wertes an sich bereits eine Information bereitstellt. Unabhängig davon müssen die fehlenden Daten bereinigt werden, da viele der Analyse-Verfahren keine fehlenden Daten akzeptieren. Dabei lassen diese sich durch verschiedene Techniken bereinigen. Es besteht zum Beispiel die Möglichkeit die fehlenden Werte manuell zu setzen, was allerdings mit zunehmender Datenmenge unpraktikabel wird. Alternativ könnten die fehlenden Werte vorhergesagt werden, wenn diese in einer Beziehung zu anderen Attributen stehen. Beispielsweise lässt sich der Wohnort einer Person durch die Postleitzahl ermitteln. Existiert keine Relation zwischen Attributen kann auch ein konstanter Wert wie *unbekannt* oder ein beliebiger numerischer Wert die fehlenden Werte ersetzen.

Falsche Daten können etwa Werte sein, die einen gegebenen Wertebereich verletzen. Beispielsweise könnte ein Wertebereich für das Alter einer Person auf einen Bereich von 0 bis 100 Jahren definiert werden. Damit wären alle Person mit einem Alter < 0 oder > 100 als fehlerhaft identifiziert. Bei falschen Daten ist eine händische Korrektur oder ein speziell dafür entwickeltes Programm kaum vermeidbar.

Syntaktische Probleme sind im Grunde eine Art falscher Daten. Hierbei handelt es sich um Schreibfehler und Abkürzungen, die in eine einheitliche Form überführt werden sollten. Als Beispiel soll die Schreibweise eines Straßennamens dienen. So könnte es für dieselbe Straße die Varianten, Abkürzungen oder Schreibfehler *Hauptstraße*, *Hauptstrasse*, *Hauptstr.*, *Huaptstraße* geben. Diese Werteausprägungen sind zu ermitteln und in eine gemeinsame Form zu überführen. Als Gegenstück zu den syntaktischen Problemen können auch **semantische Probleme** auftreten. Das heißt ein Attribut besitzt einen Wert, dessen Bedeutung eher einem anderen Attribut zugeordnet gehört. Als Beispiel sollen die Attribute *Adresse* und *E-Mail* dienen, wobei dem Attribut *Adresse* ein Wert *mustermann@mail.de* zugeordnet ist. Dieser

Wert repräsentiert allerdings eine E-Mail-Adresse und sollte demnach auch dem korrekten Attribut *E-Mail* zugeordnet werden.

Die hier beschriebenen Problematiken und Reaktionen bzw. Lösungen repräsentieren nur einen Teil dieses Themenbereiches. Ein weiterer Einblick kann durch die eingangs erwähnte Literatur erfolgen.

2.2.3. Datentransformation

Die Phase der Datentransformation hat das Ziel die Daten in eine für das verwendete Data-Mining-Verfahren benötigten Form zu bringen. Das ist nötig, da beispielsweise einige Verfahren nur mit numerischen Daten arbeiten können. Dementsprechend müssen die nicht-numerischen Daten in numerische überführt werden. Diese Phase wird demnach für einen erfolgreichen Abschluss unter Berücksichtigung des in der Folgephase verwendeten Verfahrens durchgeführt. Typisch vorzunehmende Anpassungen sind Änderungen der Datentypen, Bearbeitungen von Zeichenketten, Datumsangaben oder auch Maßeinheiten. Zudem ist es für einige Verfahren nötig, dass die Daten kodiert und normalisiert werden. Des Weiteren kann es sinnvoll sein eine Attribut-Selektion durchzuführen, da die Daten oftmals eine Vielzahl von Attributen besitzen. Im folgenden dieses Abschnittes sollen kurz die möglichen Probleme mit Zeichenketten, Datumsangaben und Maßeinheiten beschrieben werden, ehe das Hauptaugenmerk der Normalisierung, sowie der Kodierung und damit einhergehend der Anpassungen von Datentypen gilt. Anschließend wird noch ein weiterer wichtiger Teil der Datentransformation, die Attribut-Selektion, thematisiert.

Zeichenketten, Datumsangaben und Maßeinheiten: Bei Verfahren, die mit Zeichenketten arbeiten, ist zu prüfen, ob diese mit Umlauten, Leerzeichen, sowie Groß- und Kleinschreibung umgehen können, um gegebenenfalls Anpassungen an diesen vorzunehmen. Bei Datumsangaben ist darauf zu achten, dass diese ein einheitliches Format vorweisen. Daten aus unterschiedlichen Ländern können wegen der landesüblichen Datumsnotation unterschiedliche Formate wie 15. Jan 2017, 15-01-17 oder 01-15-17 aufweisen und sollten in ein einheitliches Format überführt werden. Zudem können Zeitangaben unterschiedlicher Zeitzonen entstehen, die ebenfalls anzupassen sind. Unterschiedliche Maßeinheiten wie die Angabe einer Länge in Yard und Metern, sind wie bereits in Unterabschnitt 2.2.1 erwähnt, in eine gemeinsame Einheit zu überführen. Falls möglich sollten diese Anpassungen bereits bei der Datenintegration durchgeführt werden.

Kodierung: Häufig ist auch eine Anpassung der Datentypen vorzunehmen, da das verwendete Verfahren nur bestimmte Datentypen zulässt. Einige Data-Mining-Algorithmen, wie zum Beispiel der ID3-Algorithmus, können nicht mit metrischen Daten arbeiten, weshalb dort eine Umwandlung in nominale bzw. ordinale Daten nötig ist. Eine Variante metrische Daten in nominale bzw. ordinale umzuwandeln, bietet die Kategorisierung oder auch Binnig[2] genannt. Es könnten bei einer Kategorisierung Intervalle gebildet werden, denen Bezeichnungen zugeordnet werden. Als Beispiel soll das Attribut *Gewicht kg* dienen, bei dem ein Intervall [0-60] die Bezeichnung *leicht* erhalten könnte. Existiert hingegen nur eine kleine Anzahl von Werten, so kann eine Kategorisierung durch die Umwandlung des numerischen Datentyps in einen textuellen erfolgen. Hingegen zu den Verfahren, die nicht mit metrischen Attributen umgehen können, existieren ebenso Verfahren, die ausschließlich mit metrischen Attributen arbeiten. Werden also metrische Attribute benötigt, so sind die textuellen Werte auf numerische abzubilden bzw. zu kodieren. Zu Beachten gilt es bei textuellen Werten, wie *sehr leicht*, *leicht*, *mittel* und *schwer*, dass es sich um ordinale Daten, also Daten mit Ordnungsrelation, handelt. Hierbei ist es wichtig, dass die Ordnung auch nach Kodierung der Werte erhalten bleibt. Ein mögliche Kodierung dieser Werte könnte wie folgt aussehen:

$$\begin{aligned} \textit{sehr leicht} &\rightarrow 0 \\ \textit{leicht} &\rightarrow 1 \\ \textit{mittel} &\rightarrow 2 \\ \textit{schwer} &\rightarrow 3 \end{aligned}$$

Aber auch eine alternative Kodierung wäre möglich:

$$\begin{aligned} \textit{sehr leicht} &\rightarrow 0 \\ \textit{leicht} &\rightarrow 1 \\ \textit{mittel} &\rightarrow 3 \\ \textit{schwer} &\rightarrow 4 \end{aligned}$$

Beide Kodierungen sind korrekt und lassen die Ordnung *sehr leicht* < *leicht* < *mittel* < *schwer* unangetastet. Der Unterschied dieser Kodierungsvarianten sind die Abstände zwischen den Werten *leicht* und *mittel*, wodurch bei Abstand-basierten Verfahren die Erzielung unterschiedlicher Ergebnisse möglich ist. Ebenso denkbar wäre es eine normierte Kodierung vorzunehmen, die allerdings auch nach der Kodierung erfolgen kann.

[2] Die Klasseneinteilung

Normalisierung: Die Normalisierung transformiert Werte auf ein numerisches Intervall wie etwa [0,1]. Für die Umsetzung wird der minimale $min(x_i)$ und der maximale Wert $max(x_i)$ eines Attributs benötigt. Aus diesen Werten ergibt sich der normalisierte Wert x_{new} durch folgende Berechnung:

$$x_{new} = \frac{x - min(x_i)}{max(x_i) - min(x_i)} \tag{2.1}$$

Eine Normalisierung kann nötig sein, wenn Attribute sehr hohe Werte im Vergleich zu anderen aufweisen. Durch diese hohen Werte hätte dieses Attribut einen eventuell ungewollten dominierenden Einfluss auf das Resultat. Werden mit einer Normierung im Intervall [0,1] keine zufriedenstellenden Resultate erzielt, kann für ein verbessertes Resultat das Intervall auf einen Wertebereich [0,X] angepasst werden. Der Wert *X* wird gemäß der geschätzten Bedeutung eines Attributes gewählt.

Attribut-Selektion: Oftmals sind für die Daten-Analyse nicht alle vorhandenen Attribute der Datenmenge relevant. Da eine große Anzahl von Attributen die Effizienz des verwendeten Verfahrens, sowie die Qualität des Resultats negativ beeinflussen kann, sollte im Vorwege eine Auswahl der relevanten Attribute getroffen werden. Jedoch führen nicht nur viele Attribute zu einer hohen Komplexität, sondern auch viele Attributwerte. Daher werden zunächst die Techniken der Datenreduktion behandelt, ehe die Techniken zur Auswahl geeigneter Attribute beschrieben werden.
Um viele Attributwerte auf einige wenige zu minimieren, kann die Datenkompression verwendet werden. Bei der Datenkompression steht genau der Sachverhalt der Verringerung der Anzahl von Werten im Vordergrund. Dahinter verbirgt sich das Zusammenfassen von mehreren Werten zu einem. Beispielsweise könnten die Werte für Getränke *Altbier*, *Kölsch*, *Pils* und *Weizenbier* zu einem Wert *Bier* zusammengefasst werden. Auch eine Aggregation, die das Zusammenfassen von Datensätzen oder Attributen zum Ziel hat, könnte angewandt werden. Liegen die Attribute *Straße*, *Hausnummer* und *Wohnort* vor, so wäre es denkbar diese zu einem Attribut *Adresse* zusammenzufassen. Aber nicht nur die Verdichtung von den vorhandenen Daten kann für eine Dimensionsreduktion eingesetzt werden. Es gibt ebenfalls Techniken zur Identifizierung der für die Analyse relevanten Attribute. Eine Attribut-Selektion könnte etwa manuell vorgenommen werden, wenn genügend Anwendungswissen über die Bedeutung der Attribute vorhanden ist. Jedoch ist dieses Wissen oftmals nicht vorhanden, zudem eine manuelle Selektion mit hohem Aufwand verbunden ist. Demgegenüber stehen die Methoden zur automatisierten Auswahl der relevanten Merkmale. Grundsätzlich wird dabei in die zwei Klassen der *Wrapper* und *Filter* unterschieden. Ein Wrapper führt auf jedem Unterraum der

Merkmale ein Data-Mining-Verfahren aus und bewertet das entstandene Resultat nach einem gewissen Kriterium. Da der Aufwand eines optimalen Wrapper-Algorithmus, der jede mögliche Teilmenge der Attributmenge berücksichtigt, zu hoch wäre, gibt es alternative Ansätze. Meist umfasst dabei die Attributmenge anfangs entweder die leere Menge oder eben die Gesamtmenge der Attribute. Je nach gewählter Anfangsmenge wird sukzessiv das Attribut hinzugefügt bzw. entfernt, das für das verwendete Data-Mining-Verfahren am besten oder schlechtesten bewertet wurde. Dies wird bis zur Erreichung der besten Attributmenge durchgeführt. Ein Filter hingegen nutzt ein Bewertungskriterium, welches direkt auf die Attributmenge angewandt wird und ohne den Einsatz eines Data-Mining-Verfahrens auskommt. Als Bewertungskriterien können unter anderem statistische Verfahren, wie die Varianzberechnung oder Verfahren zur Berechnung eines Korrelationskoeffizienten, genutzt werden. Die Unterschiede dieser zwei Methoden zur Attribut-Selektion liegen in der Ausführungsgeschwindigkeit und der Qualität. Der Filter hat gegenüber dem Wrapper einen Geschwindigkeitsvorteil, jedoch ist die Qualität der Attributs-Selektion beim Wrapper höher, da diese auf den später angewendeten Algorithmus optimiert werden kann. Eine konkrete Umsetzung bzw. Verwendung der Techniken, sowie deren Nutzen ist in dem Unterabschnitt 3.4.2 beschrieben.
Neben den hier beschriebenen Problemen, Techniken und Verfahren sind weitere der Literatur zu entnehmen. Nach erfolgreichem Abschluss der Phase der Datentransformation liegen die Daten in der für das Data-Mining-Verfahren benötigten Form vor.

2.2.4. Data Mining

In der Data-Mining Phase kommt nun ein ausgewähltes Verfahren zur Analyse der vorbereiteten Daten zum Einsatz. Olaf Herden hat das Data-Mining wie folgt definiert:

> Data Mining bezeichnet die Auswertung vorhandener Daten mit dem Ziel, bisher nicht explizit hergestellte Zusammenhänge offenzulegen (Knowledge Discovery).
>
> Herden (2015)

Für die Auswahl eines geeigneten Verfahrens, muss zunächst bestimmt werden, welche grundlegende Data-Mining-Aufgabe vorliegt. Dabei kann etwa in Cluster-Analyse, Klassifikation und Assoziationsanalyse, um nur einige zu nennen, unterschieden werden. Eine typische Cluster-Analyse Aufgabe besteht darin eine Datenmenge in Gruppen ähnlicher Objekte einzuteilen, um diesen eine Bezeichnung zuzuordnen und somit Klassen zu bilden. Beispielsweise können Kundendaten analysiert werden und Gruppierungen für die Zahlungsfähigkeit der Kunden gebildet werden. Liegen die Attributwerte samt Klassifizierung vor, so kann eine Funktion gelernt werden, die zukünftige Kunden anhand ihrer Attributwerte in eine der Klassen

einordnet und damit eine Vorhersage über die Zahlungsfähigkeit eines Kunden trifft. Dieser Fall beschreibt ein typisches Problem der Klassifizierung. Bei der Assoziationsanalyse handelt es sich um die Analyse häufig auftretender und starker Zusammenhänge in den Daten. Ein wohl bekanntes Beispiel liefern die *Kunden, die diesen Artikel gekauft haben, kauften auch* Anzeigen auf Online-Shopping Plattformen. Wurde die Anwendungsklasse der Aufgabenstellung identifiziert, ist im Anschluss ein geeignetes Verfahren für die Durchführung zu wählen. Da in dieser Arbeit eine Aufgabenstellung für eine Cluster-Analyse vorliegt, richtet sich der Fokus eben auf diese Anwendungsklasse. Für andere Anwendungsklassen sei nochmals auf die Literatur von (Cleve und Lämmel, 2016) verwiesen.
Eine Cluster-Analyse hat, unabhängig vom verwendeten Verfahren, das Ziel Gruppen von Objekten zu finden, die sich möglichst ähnlich sind. Das heißt wiederum, dass Objekte unterschiedlicher Gruppen sich möglichst unähnlich sind. Bei einer Cluster-Analyse handelt es sich um *unüberwachtes* Lernen, das heißt die zu entdeckenden Muster sind im Vorhinein nicht bekannt. Um Gruppen ähnlicher Objekte zu bilden, nutzen Cluster-Analyse-Verfahren Abstands- oder Ähnlichkeitsmaße, wie beispielsweise die *euklidische Distanz*. Das Abstandsmaß bietet im Grunde genommen gleichzeitig ein Ähnlichkeitsmaß, denn umso höher die Distanz zweier Objekte ist, desto unähnlicher sind sich diese. Bei einer Distanz von 0 oder nahe 0, liegen identische oder sehr ähnliche Objekte vor. Handelt es sich bei den zu analysierenden Objekten um mehrdimensionale Objekte, so lässt sich die Ähnlichkeit über die Summe der Einzeldistanzen ermitteln.
Die Cluster-Analyse lässt sich unter anderem in die folgenden 3 Verfahren zur Clusterbildung unterscheiden:

- Partitionierende Clusterbildung
- Hierarchische Clusterbildung
- Dichtebasierte Clusterbildung

Die Gemeinsamkeiten von Cluster-Analyse-Verfahren wurden bereits beschrieben. Im Folgenden werden die Arbeitsweisen der aufgeführten Verfahren etwas näher gebracht, sowie eine Bewertung dieser vorgenommen. Für einen tieferen Einstieg in diese und andere Techniken zur Clusterbildung kann zur bereits genannten Literatur auch das Buch Kaufman und Rousseeuw (2008) empfohlen werden.

Partitionierende Clusterbildung

Die partitionierende Clusterbildung zerlegt die Datenmenge in k zufällige Anfangspartitionen und ordnet die Objekte iterativ zwischen den Clustern so um, dass sich die Güte der Cluster über die Minimierung einer Fehlerfunktion stets verbessert. Die Cluster können durch einen Mittelwert bzw. Mittelpunkt der ihm zugeordneten Objekten, den sogenannten Centroid, repräsentiert werden. Eine weitere Möglichkeit bietet die Cluster-Repräsentation über den Medoid, einem typischen Vertreter des Clusters. Die Wahl eines Medoid kann etwa auf das Objekt fallen, das in dem geringsten Abstand zum Centroid liegt. Ebenfalls existieren weitere Möglichkeiten zur Repräsentation von Clustern, die hier nicht behandelt werden. Wie eingangs erwähnt, werden die Objekte iterativ dem Cluster, dessen Repräsentant sie am nächsten sind, zugeordnet. Mit jedem Durchlauf wird nicht nur die Cluster-Zugehörigkeit der Objekte berechnet, sondern auch eine Neuermittlung des Cluster-Repräsentanten durchgeführt. Abgeschlossen ist die partitionierende Clusterbildung, wenn keines der Objekte einem anderen Cluster zugeordnet wird. Die durch ein partitionierendes Verfahren gebildeten Cluster erfüllen die Anforderungen, dass ein Cluster aus mindestens einem Objekt besteht, sowie jedes Objekt genau einem Cluster angehört. Diese Anforderungen implizieren, dass es nicht mehr Cluster als Objekte geben kann. Die wohl bekanntesten Verfahren der partitionierenden Clusterbildung sind der oft als Referenzverfahren erwähnte *k-Means* und der *k-Medoid* Algorithmus, die der Klasse der Mittelpunkt-Verfahren angehören.
Problematisch stellt sich die Ermittlung der Anzahl der zu bildenden Cluster k dar, denn diese ist im Vorhinein meist nicht bekannt. Ebenso haben partitionierende Verfahren oftmals Probleme mit der Erkennung von Ausreißern. Zudem ist die Bildung von nicht konvexen Clustern[3] mit diesen Verfahren nicht möglich. Im Gegensatz zu anderen Verfahren ermöglicht diese Vorgehensweise den Objekten den Wechsel des Clusters, wenn diese einem anderen neu berechneten Cluster-Repräsentanten näher als dem eigenen sind.

Hierarchische Clusterbildung

Bei der hierarchischen Clusterbildung wird eine Baumstruktur erstellt, bei der die Knoten mit einer minimalen Distanz zueinander verschmolzen werden. Dabei repräsentiert die *Wurzel* des Baumes die gesamte Datenmenge. Die inneren *Knoten* stehen für ein Cluster, in denen alle unter diesem liegenden Teilbäume vereinigt sind. Die einzelnen Objekte werden durch die *Blätter* dargestellt. Dabei lässt sich in zwei hierarchische Techniken unterscheiden, die *agglomerative* und die *divise*. Deren Vorgehensweisen unterscheiden sich in der Richtung des

[3] Schaubild im Anhang A.2

Aufbaus der Baumstruktur. Die agglomerative Clusterbildung beginnt mit der größtmöglichen Anzahl von Clustern, das heißt jedes Objekt repräsentiert ein Cluster. Die zwei Cluster, die sich am nächsten sind, werden auf einer Ebene höher zu einem Cluster vereinigt, bis nur noch ein Cluster existiert. Die divise Clusterbildung geht genau entgegengesetzt vor und startet mit dem initialen Cluster, dass die gesamte Objektmenge enthält. Die Cluster werden dann auf den tieferen Ebenen aufgeteilt, bis jedes Cluster genau ein Objekt repräsentiert.
Bei der hierarchischen Clusterbildung handelt es sich um ein bereits erwähntes Verfahren, bei dem ein Objekt nach Zuordnung eines Clusters keinen Wechsel vornehmen kann. Eine Vorgabe der Anzahl der gesuchten Cluster entfällt hingegen bei hierarchischen Verfahren. Allerdings sind sie, im Gegensatz zu anderen Verfahren, nur auf eine vergleichsweise geringe Menge von Objekte anwendbar. Ähnlich wie bei der partitionierenden Clusterbildung besteht auch hier das Problem der Ausreißer-Erkennung.

Dichtebasierte Clusterbildung

Die dichtebasierte Clusterbildung setzt da an, wo die partitionierende Clusterbildung Schwierigkeiten hat und zwar bei der Bildung von nicht konvexen Clustern. Als nicht konvexe Cluster lassen sich jene bezeichnen, die sich wie ein Schlauch durch den n-dimensionalen Raum ziehen. Bei der dichtebasierten Clusterbildung bilden die Cluster eine Gruppe von Objekten, die in einer bestimmten Dichte zueinander stehen. Die unterschiedlichen Cluster sind durch Regionen geringerer Dichte voneinander getrennt. Ein Cluster wächst demnach solange die Dichte von Objekten in der Nachbarschaft einen Schwellwert überschreitet. Ein in der Literatur oft genanntes Verfahren ist der *DBScan*-Algortihmus.
Wie bereits erwähnt, finden dichtebasierte Verfahren nicht konvexe Cluster. Zudem ist eine Angabe der Anzahl der gesuchten Cluster k bei diesem Vorgehen nicht nötig. Ebenfalls hat dieses Verfahren im Gegensatz zu den bisherigen kein Problem mit der Erkennung von Ausreißern. Als Problem hingegen ist die optimale Wahl der Parameter zur Bestimmung der Dichte anzusehen.

2.2.5. Evaluation

In der letzten Phase werden die durch das Data-Mining erzielten Resultate von einem Experten in Bezug auf die definierten Ziele evaluiert. Dazu wird eine Bewertung hinsichtlich der *Gültigkeit*, *Neuartigkeit*, *Nützlichkeit* und *Verständlichkeit* des Ergebnisses durchgeführt. Diese Bewertung kann unabhängig vom verwendeten Verfahren angewandt werden. Im Folgenden sollen die Bewertungskriterien kurz erläutert werden.

- Die Gültigkeit beschreibt eine Art Wahrscheinlichkeit mit der die gefundenen Muster auch auf neue Daten zutreffen.
- Das Kriterium der Neuartigkeit bewertet, ob das Analyse-Ergebnis die bisherigen Kenntnisse erweitert oder gänzlich neue liefert.
- Der Aspekt der Nützlichkeit bewertet, ob das ermittelte neue Wissen auch praktisch verwertbar und anwendbar ist.
- Die Verständlichkeit ist ein Maß dafür, wie gut ein Resultat vom Experten nachvollzogen werden kann.

Liegt ein erwartetes Resultat vor, so scheint eine Bewertung trivial. Bei einer Durchführung einer Klassifikation, einem Verfahren des überwachten Lernens, kann also anhand der vorliegenden bereits klassifizierten Daten überprüft werden, ob ein neues Objekte korrekt eingeordnet wurde. Im Folgenden wird sich aber speziell auf die Bewertung von Cluster-Analysen konzentriert. Für einen Einblick in die Bewertungsmaße anderer Anwendungsklassen empfiehlt sich die erwähnte Literatur.
Liegt ein Resultat einer Cluster-Analyse vor, kann im Regelfall keine Überprüfung einer korrekten Zuordnung durchgeführt werden, da die Zielcluster eben nicht bekannt sind. Dadurch stellt sich auch der Vergleich zweier erzielter Resultate als schwierig dar. Um eine Vergleichbarkeit zu schaffen, kann die Qualität der gebildeten Cluster durch verschiedene Herangehensweisen ermittelt werden.

Variante 1: Es ist gefordert, dass sich die Objekte innerhalb eines Clusters möglichst ähnlich sind. Es werden daher für jedes Objekt eines Clusters die Abstände bzw. die Abweichungen zum jeweiligen Cluster-Repräsentanten ermittelt und aufsummiert. Die ermittelten Summen werden dann nochmals für alle Cluster aufsummiert.

$$G_1 = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, m_i)^2 \tag{2.2}$$

Der Wert m_i steht für den Cluster-Repräsentanten vom Cluster C_i. Umso geringer der Wert G_1 ist, desto besser ist die Qualität der Cluster zu bewerten. Dementsprechend ist das Ergebnis zweier Resultate mit dem kleineren Wert als besser zu bewerten.

Variante 2: Es ist gefordert, dass die Cluster möglichst weit voneinander entfernt liegen. Hier können beispielsweise die Distanzen der Cluster-Repräsentanten aufsummiert werden.

$$G_2 = \sum_{1 \leq i \leq j \leq k} dist(m_j, m_i)^2 \tag{2.3}$$

Das Resultat mit den größeren Abständen der Cluster G_2 ist als besser zu bewerten.
Die betrachteten Varianten gehen allerdings von einer fixen Anzahl der Cluster k aus. Mit einer variablen Anzahl von Clustern sind diese Varianten ungeeignet, da das optimale Resultat ein k = *Anzahl von Objekten* ergeben würde. Eine Möglichkeit mit variablen k umzugehen ist, Cluster-Analysen für verschiedene k durchzuführen und die daraus entstandenen Clusterbildungen hinsichtlich ihrer Qualität zu vergleichen. Eine weitere Möglichkeit ergibt sich durch den *Silhouetten*-Koeffizienten. Der Silhouetten Koeffizient berechnet sich aus dem durchschnittlichen Abstand $dist_1(x)$ eines Objektes x zu den Objekten desselben Clusters, sowie dem durchschnittlichen Abstand $dist_2(x)$ zu den Objekten des nächstgelegenen Clusters.

$$s(x) = \frac{dist_1(x) - dist_2(x)}{max(dist_1(x) - dist_2(x))} \tag{2.4}$$

Der Koeffizient eines Clusters ergibt sich letztlich aus dem Mittelwert der Koeffizienten aller Objekte eines Clusters. Dieser kann einen Wert zwischen -1 und 1 annehmen. Aus dem Koeffizienten erschließen sich die folgen Aussagen:

0.71 - 1.0 :	Eine Starke Struktur gefunden.
0.51 - 0.7 :	Eine brauchbare Struktur gefunden.
0.26 - 0.5 :	Eine schwache Struktur gefunden.
< 0.25 :	Keine wesentliche Struktur gefunden.

Tabelle 2.5.: Bewertung des Silhouetten Koeffizienten (Kaufman und Rousseeuw, 2008)

Diese Bewertung der Cluster stellt aber nur ein theoretisches Maß für qualitativ gute Cluster dar. Dieser kann beispielsweise für einen Vergleich von Clustering-Verfahren eingesetzt werden. Ebenfalls ist dieser Koeffizient durchaus für Experimente, wie etwa zur Findung einer geeigneten Konfiguration eines Verfahrens, einsetzbar. Über die Sinnhaftigkeit eines Resultats kann dieser Koeffizient jedoch keine Aussage treffen. Deshalb ist es notwendig, dass eine Bewertung in Zusammenarbeit mit einem Fachexperten durchgeführt wird, denn nur dieser kann aufgrund seiner Fachkenntnisse eine Bewertung hinsichtlich der zu Anfang dieses Abschnittes aufgeführten Kriterien durchführen. Fällt die Bewertung bezüglich dieser Kriterien negativ aus, das heißt das Resultat ist beispielsweise nicht verständlich oder die extrahierten Informationen

sind nicht nützlich, so kann in eine frühere Phase des KDD-Prozesses zurückgekehrt werden. Es könnte beispielsweise ein Rücksprung in die Phase des Data-Mining durchgeführt werden, um die Parameter des verwendeten Verfahrens anzupassen. Ebenso denkbar wäre ein Rücksprung in die Datentransformation, um Änderungen an der Attribut-Selektion vorzunehmen, da die bisherige Auswahl nicht zur Erreichung der Ziele geeignet war.

2.3. Aufgabenstellung

Es soll Stück für Stück ein Teilbereich aus dem Datenbestand der Teile selektiert werden und der KDD-Prozess im Weiteren durchlaufen werden. Dafür ist zunächst ein Verständnis für die Anwendung und die Daten aufzubauen. Ist die Datenselektion vorgenommen, folgt die Integration in einen zentralen Datenbestand. Ebenfalls ist die Qualität der Daten zu prüfen und auf die Eignung für das verwendete Data-Mining-Verfahren zu untersuchen. Gegebenenfalls sind Maßnahmen zur Steigerung der Qualität des Datenbestandes, sowie die Überführung in die für das Data-Mining-Verfahren benötigte Struktur nötig. Im Zuge der Datentransformation soll durch den Einsatz verschiedener Verfahren zur Attribut-Selektion die Qualität und die Eignung dieser Verfahren untersucht werden. Sind die Daten für das Data-Mining vorbereitet, ist neben der Überprüfung verschiedener Clustering-Verfahren hinsichtlich der Eignung zur Erstellung eines Clusterings im Kontext technischer Artefakte auch die Wahl geeigneter Parameter der Verfahren zu treffen. Ist die Cluster-Analyse durchgeführt, gilt es die Resultate einem Fachgebietsexperten zu präsentieren und diesen in Bezug auf die Art der Gruppierungen zu sensibilisieren. Im Falle eines unbrauchbaren Resultats, sind die Gründe zu ermitteln und weitere Maßnahmen oder Anpassungen an einer früheren Phase des KDD-Prozesses vorzunehmen, sowie ein weiterer Durchlauf zu starten. In den Kapiteln 3 und 4 folgt die Umsetzung dieser Aufgabenstellung anhand dem Teilbereich der Schrauben.

3. Praktische Durchführung

Für die praktische Durchführung der in Abschnitt 2.3 beschriebenen Aufgabenstellung wurde eine Toolchain zur Umsetzung des in Abschnitt 2.2 beschriebenen KDD-Prozesses aufgebaut. Diese Toolchain ist Abbildung 3.1 dargestellt und umfasst neben den Tools auch deren Verarbeitungsschritte.

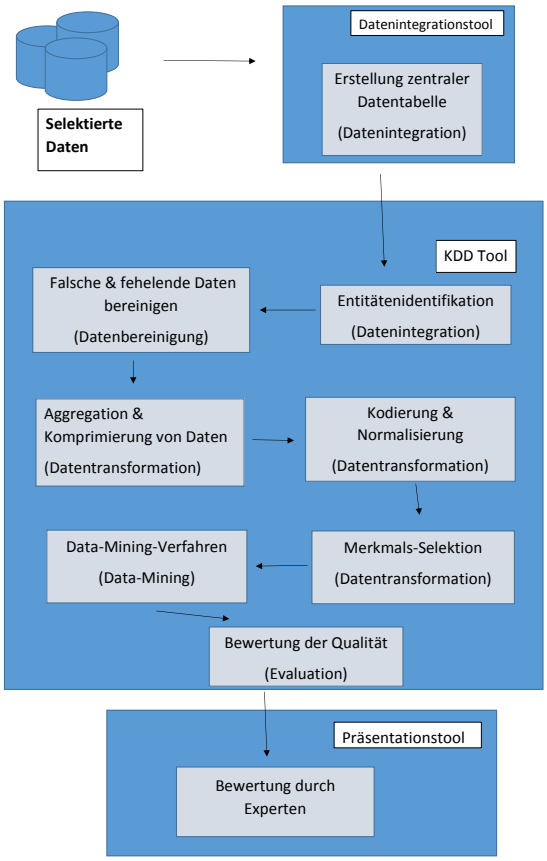


Abbildung 3.1.: Toolchain inklusive derer Verarbeitungsschritte

Dieses Kapitel behandelt zunächst die Auswahl geeigneter Programme zur Realisierung der Toolchain. Anschließend wird im Abschnitt 2.2 die praktische Durchführung beschrieben. Dafür werden die in den einzelnen KDD-Phasen aufgetretenen Problemen erläutert, sowie eine Lösung für diese präsentiert.

3.1. Verwendete Programme

Zur Umsetzung des KDD-Prozesses wurde nach einer nicht kommerziellen Software-Lösung gesucht, die bereits viele Lösungen für bekannte Probleme mitbringt. Dabei wurden die Programme RapidMiner und KNIME, sowie die Programmiersprache R bei der Auswahl einer geeigneten Software berücksichtigt, da dieses die gängigsten Open-Source Software Produkte zur Datenanalyse sind.
RapidMiner ist eine in Java realisierte und für Windows, Mac und Linux verfügbare Software für maschinelles Lernen, sowie Data Mining. 2001 wurde die Software ursprünglich unter dem Namen YALE entwickelt, ehe sie 2007 in RapidMiner umbenannt wurde. Seit 2004 ist RapidMiner auf *SourceForge*, einem Filehostingdienst für Softwareprojekte, gehostet und bereits seit einigen Jahren auf dem Markt. Es wird eine grafische Oberfläche zur Erstellung von analytischen Workflows geboten, wofür über 1500 Operatoren, unter anderem für Ein- und Ausgabe, Datenvorverarbeitung, Data-Mining, Text-Mining, Web-Mining und Visualisierung, bereitgestellt werden. Ebenfalls lassen sich die Programmiersprachen R und Python, sowie Weka [1] in RapidMiner integrieren. Einen FAQ bietet RapidMiner Studio nicht, dafür aber eine Community, die für Fragen kontaktiert werden kann. In der kostenfreien Version ist RapidMiner auf 10.000 Datensätze, sowie einen logischen Prozess limitiert, wodurch die Nutzung zeitaufwendig und nur eingeschränkt möglich ist.
KNIME ist eine ebenfalls in Java entwickelte und unter Windows, Mac und Linux verfügbare Software zur Durchführung von Datenanalysen. Die Entwicklung begann 2004 bis dann 2006 die erste Version veröffentlicht wurde. Auch KNIME wird über eine Benutzeroberfläche bedient, über die, ähnlich wie in RapidMiner, Workflows erstellt werden können. Die Workflows basieren auf dem Pipelining-Konzept. Das Pipelining-Konzept besagt, dass das Ergebnis oder die Ausgabe eines Verarbeitungsschrittes die Eingabe des darauffolgenden Schrittes ist. KNIME bietet über 1000 Module für Datenintegration, Datentvorverarbeitung, Datenanalyse und -visualisierung, sowie die Integration von Programmen in R, Python und Java. Ebenfalls besteht die Möglichkeit zur Erweiterung mit Weka und anderer Tools. KNIME bietet einen FAQ, sowie einen guten Support durch die KNIME-Community. Eine Limitation in Sachen

[1] Softwaretool, das verschiedene Techniken für den KDD Prozess bereitstellt

Datensätze existiert im Gegensatz zu RapidMiner nicht.
Eine weitere Alternative bietet die Programmiersprache R für statistische Berechnungen und Grafiken. R ist unter Windows, Mac und UNIX Plattformen verfügbar. Die Programmiersprache wurde 1992 entwickelt und seitdem stetig gewachsen. Der Funktionsumfang von R kann durch eine Vielzahl von zusätzlichen Paketen erweitert werden, darunter auch Miningverfahren. R bietet auch Schnittstellen zu anderen Programmiersprachen, wie beispielsweise Java oder Python. Es existieren ein FAQ, sowie eine Community und eine Menge von Blogs, sowie Foren die R thematisieren.
Jede dieser Software Lösungen umfasst mehrere Anwendungsklassen des Data-Mining und bietet somit nicht nur eine Lösung für ein bestimmtes Problem. Da es sich nicht um Big Data handelt, werden für akzeptable Laufzeiten keine Spezialrechner benötigt. Aufgrund der Tatsache, dass KNIME und RapidMiner über eine Benutzeroberfläche bedienbar sind, sowie R in beide Programme integrierbar ist, wurde sich gegen eine reine Umsetzung in R entschieden. Jedoch wird auf R nicht gänzlich verzichtet, da R unter anderem viele unterschiedliche Verfahren verschiedener Miningtechniken zur Verfügung stellt. KNIME und RapidMiner sind beides etablierte Software-Lösungen, die sich in ihrer Funktionalität sehr ähneln. Letztendlich fiel die Wahl auf KNIME als KDD-Plattform, da die Einbindung von Java Programmen durch vorhandene Module sehr einfach ist. Zudem ist KNIME nicht in der Anzahl der zu verarbeitenden Datensätze limitiert. Bei der Datenintegration wurde der Teil des Überführen der Daten in eine Datentabelle in der Programmiersprache Visual Basic realisiert, da der Umgang mit dieser Sprache vertraut ist, könnte jedoch mit jeder beliebigen Programmiersprache realisiert werden. In KNIME ist dieser Vorgang leider nicht umsetzbar, da es nicht möglich ist, Tabellen mit unterschiedlichen Tabellendefinitionen automatisiert einzulesen. Zur Präsentation der Ergebnisse wird Microsoft Excel genutzt.

3.2. Datenselektion und -integration

Die Datenselektion wurde von der Firma Claudius Peters vorgenommen und umfasst zunächst den Gegenstandsbereich der Schrauben. Zur Verfügung gestellt sind die Daten in Form von 55 csv-Dateien. Da die selektierten Dateien zum Teil unterschiedliche Tabellendefinitionen aufweisen, werden diese zunächst zu einer gemeinsamen Datentabelle zusammengefasst. Wie bereits erwähnt, lässt die KDD-Plattform KNIME das Einlesen von Tabellen mit unterschiedlichen Tabellendefinitionen nicht zu. Aufgrund dessen erfolgt das Zusammenführen der unterschiedlichen Datentabellen in einen zentralen Datenbestand durch ein eigen-entwickeltes Programm in der Sprache Visual Basic. Der Algorithmus 3.1 zeigt einen Pseudo Code, der die zentrale

Datentabelle erzeugt. Dort wird in Zeile 2 zunächst eine leere Datentabelle erzeugt. Der Algorithmus iteriert über die Dateien des ausgewählten Verzeichnisses und fügt die Datentabellen hinzu. Dazu dient die Operation in Zeile 5. Dort wird an die zentrale Datentabelle mit jeder Iteration eine weitere Datentabelle angefügt. Durch die Option *unionOfColumns=TRUE* werden bereits in der zentralen Datentabelle vorhandene Attribute nicht nochmal angefügt, sondern die Werte der Datentabelle *appenDataTable* entsprechend den Attributen zugeordnet.

Algorithm 3.1 Pseudocode: Datenintegration

1: $dir \leftarrow chooseDirectory()$
2: $dataTable \leftarrow newDataTable$
3: **for all** $file \in dir$ **do**
4: $\quad appendDataTable \leftarrow open(file)$
5: $\quad Concatenate(dataTable, appendDataTable, unionOfColumns = TRUE)$
6: **end for**

Nach Ausführung des Algorithmus liegt eine zentrale Datentabelle vor, die problemlos in KNIME eingelesen werden kann. Die darin enthaltenen Merkmale inklusive ihrer Beschreibungen können im Anhang A.1 eingesehen werden. Die nun vorliegende zentrale Datentabelle wurde anschließend hinsichtlich semantisch äquivalenter Merkmale untersucht. Das Merkmal *OBFBEH alpha* beschreibt laut Metadaten die Oberfläche einer Schraube, wohingegen das Merkmal *OBFSTS alpha* die Oberfläche des speziellen Schraubentyps der Steinschraube beschreibt. Da es sich bei einer Steinschraube ebenfalls um eine Schraube handelt, wurden diese Merkmale nach Rücksprache mit einem Fachexperten als äquivalent angesehen. Liegen semantisch äquivalente Merkmale vor, so sind diese zu einem Merkmal zusammenzufassen. Dabei ist darauf zu achten, dass falls beide Merkmale einen Wert besitzen, diese nicht im Widerspruch zueinander stehen. Falls die Merkmalswerte im Widerspruch zueinander stehen, wurde der Wert des Merkmals *OBFSTS alpha* als der korrekte ermittelt. Umgesetzt wurde das Zusammenfügen von Merkmalen mit dem Workflow in Abbildung 3.2.

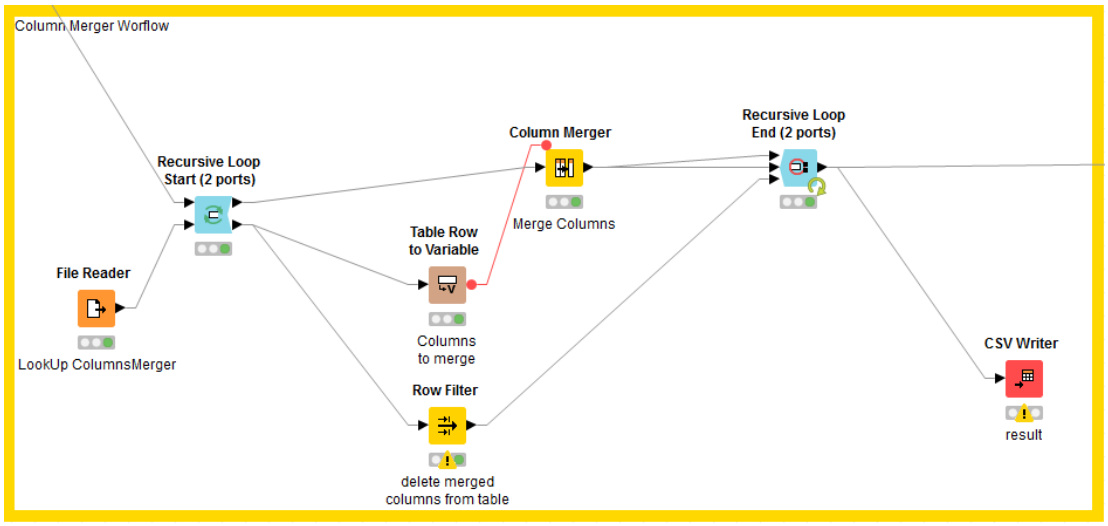


Abbildung 3.2.: Workflow: Merkmale zusammenfassen

Der Workflow arbeitet auf Basis einer Tabelle dessen Aufbau in Tabelle 3.1 zu sehen ist. Es werden die zusammenzuführenden Merkmale aus der Tabelle eingelesen und das Merkmal *SecondaryFeature* in das Merkmal *PrimaryFeature* integriert. Über den *Column-Merger-Node* kann das Merkmal mit den korrekten Werten im Falle eines Widerspruchs gewählt werden. Die Tabelle 3.1 enthält die Merkmale, die als äquivalent identifiziert und zusammengefasst wurden.

SecondaryFeature	**PrimaryFeature**
OBFBEH alpha	OBFSTS alpha
NENNLA numer.	LANGE numer.
DURCHM numer.	GEWDUR alpha
...	...

Tabelle 3.1.: Merkmale zusammenfassen

Nach Ausführung des KNIME-Workflows liegt eine zentrale Datentabelle bestehend aus den folgenden 41 Merkmalen vor:

Teile-Nr	Teilebezeichnung	GEWDUR alpha	LANGE numer.
FESTWE alpha	STOFNR alpha	GEWLGE numer.	SCHLUW numer.
KOPFHH numer.	ABVOST alpha	SICHNA alpha	TEAB01 alpha
TEAB02 alpha	KOPFBR numer.	ANSKUP alpha	LIEUMF alpha
GEWEN1 alpha	GEWLG1 numer.	GEWEN2 alpha	GEWLG2 numer.
FORM alpha	WERKST alpha	KOPFBM numer.	OBFSTS alpha
PRODKL alpha	DUSCHF numer.	NORM alpha	WERKSH alpha
HERSTE alpha	SCHABR numer.	VERPEH alpha	SCHLFO alpha
FRM962 alpha	AEENDE alpha	FESTKL alpha	ANMUTE numer.
TELSCH alpha	TECLIF alpha	DUSCHB alpha	KKLASS alpha
AUSSCR alpha			

3.3. Datenvorverarbeitung und -bereinigung

Wie bereits in Unterabschnitt 2.2.2 erwähnt, liegen die Zieldaten meist nicht in gewünschter Qualität vor. In diesem Fall spiegelt sich die mangelnde Qualität besonders durch fehlende Daten, aber auch durch Ausreißer bzw. falsche Daten wider. Die Probleme, wie beispielsweise falsche Daten im Sinne von Schreibfehlern und uneinheitlichen Abkürzungen, sind der manuellen Dateneingabe geschuldet.

3.3.1. Falsche Daten

Nach Abschluss der Datenintegration wurden die Daten zunächst auf unzulässige Werte bzw. Fehleinträge analysiert. Dabei ist aufgefallen, dass sich in einigen Merkmalen die Semantik der Werte vermischt. Beispielsweise beschreibt das Merkmal *FESTWE alpha* die Festigkeit/Werkstoff einer Schraube, enthält aber Werte einer Werkstoff-Nummer für die das Merkmal *STOFNR alpha* vorhergesehen ist. Zudem existiert ein Merkmal *WERKST alpha*, welches für den hinter einer Werkstoff-Nr. verborgenen Werkstoff bestimmt ist. Es ist also zunächst eine semantisch korrekte Zuweisung von Werten zu Merkmalen durchzuführen. Damit dies gelingt, wurden reguläre Ausdrücke auf Basis von Falk u. a. (2016) für die Identifizierung der Semantik der Merkmalswerte entwickelt. Auf Basis dieser regulären Ausdrücke wurde, speziell für diesen Anwendungszweck, Java-Programm entwickelt, das die semantisch korrekte Zuordnung von Merkmalswerten zu Merkmalen durchführt. In Abbildung 3.3 ist der Workflow mit integriertem Java-Programm zur Bereinigung der semantisch falsch zugeordneten Daten dargestellt.

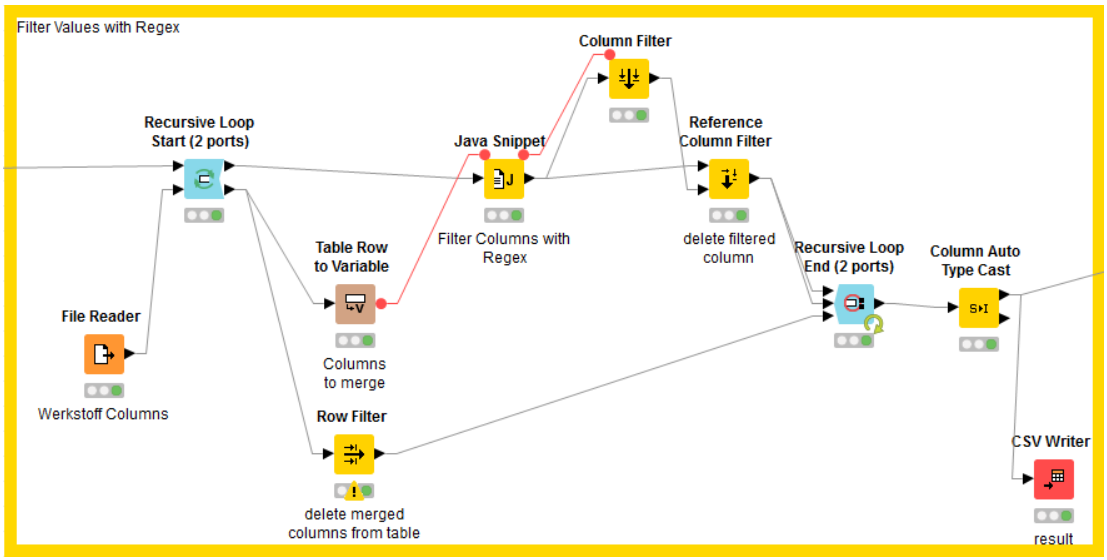


Abbildung 3.3.: Workflow: Bereinigung von falschen Daten auf Basis von regulären Ausdrücken

Für eine erfolgreiche Durchführung wurden zunächst die drei Merkmale *Werkstoff*, *WerkstoffNr* und *Festigkeit* generiert. Das sorgt dafür, dass jedes der betroffenen Merkmale analysiert werden kann, ohne das einer der falsch zugeordneten Werte überschrieben wird, ehe dieser selbst analysiert wurde. Der dargestellte Workflow liest zunächst über eine Text-Datei die zu analysierenden Merkmale ein. Der Aufbau der Text-Datei ist in der Tabelle 3.2 dargestellt. Nach Ausführung des Java-Programmes kann das analysierte Merkmal aus der Datentabelle entfernt werden, da diese Informationen nun einem der Merkmale *Werkstoff*, *WerkstoffNr* oder *Festigkeit* zugeordnet wurde.

CheckFeature
FESTWE alpha
STOFNR alpha
WERKST alpha
WERKSH alpha
FESTKL alpha

Tabelle 3.2.: Merkmale zur Werteprüfung mittels regulärer Ausdrücke

Nach Ausführung dieses Workflows liegt eine um zwei Merkmale verringerte Datentabelle vor, da mehrere Merkmale semantisch äquivalente Inhalte enthielten. Bei einer bereits im Vorhinein semantisch korrekten Zuordnung dieser Werte, hätten diese Merkmale schon in der

Phase der Datenintegration behandelt werden können. Daraus ergibt sich eine Datentabelle mit den folgenden Merkmalen:

Teile-Nr	Teilebezeichnung	GEWDUR alpha	LANGE numer.
GEWLGE numer.	SCHLUW numer.	KOPFHH numer.	ABVOST alpha
SICHNA alpha	TEAB01 alpha	TEAB02 alpha	KOPFBR numer.
ANSKUP alpha	LIEUMF alpha	GEWEN1 alpha	GEWLG1 numer.
GEWEN2 alpha	GEWLG2 numer.	FORM alpha	KOPFBM numer.
OBFSTS alpha	PRODKL alpha	DUSCHF numer.	NORM alpha
HERSTE alpha	SCHABR numer.	VERPEH alpha	SCHLFO alpha
FRM962 alpha	AEENDE alpha	ANMUTE numer.	TELSCH alpha
TECLIF alpha	DUSCHB alpha	KKLASS alpha	AUSSCR alpha
Festigkeit	WerkstoffNr	Werkstoff	

Ebenfalls unter dem Punkt der falschen Daten zu führen sind syntaktische Probleme, wie Schreibfehler, Abkürzungen oder uneinheitliche Schreibweisen. Die vorliegenden Daten weisen vorwiegend uneinheitliche Schreibweisen bzw. Abkürzungen auf. So existieren für das Merkmal *OBFSTS alpha* die vier Ausprägungen *feuerverzinkt*, *feuer_verz.*, *feuer.verz.* und *feu.verz.*. All diese verschiedenen Ausprägungen sind semantisch äquivalente Einträge, die für die Datenanalyse auf einen gemeinsamen Wert zu überführen sind. Geschieht diese Überführung nicht, werden die verschiedenen Ausprägungen für die Analyse auch als unterschiedlich angesehen und verfälschen womöglich das Ergebnis. Die Übersetzung semantisch äquivalenter Werte auf einen ebenfalls syntaktisch äquivalenten Wert erfolgt auf Basis einer Übersetzungstabelle, wie in Tabelle 3.3 dargestellt.

currentValue	**newValue**
feu.verz.	feuerverzinkt
feuer_verz.	feuerverzinkt
feuer.verz.	feuerverzinkt
...	...

Tabelle 3.3.: Werte Übersetzung

Der in Abbildung 3.4 dargestellte Workflow liest eine solche Übersetzungstabelle ein und führt über den *Cell-Replacer*-Node die Übersetzung vom Wert der Spalte *currentValue* in den Wert der Spalte *newValue* durch.

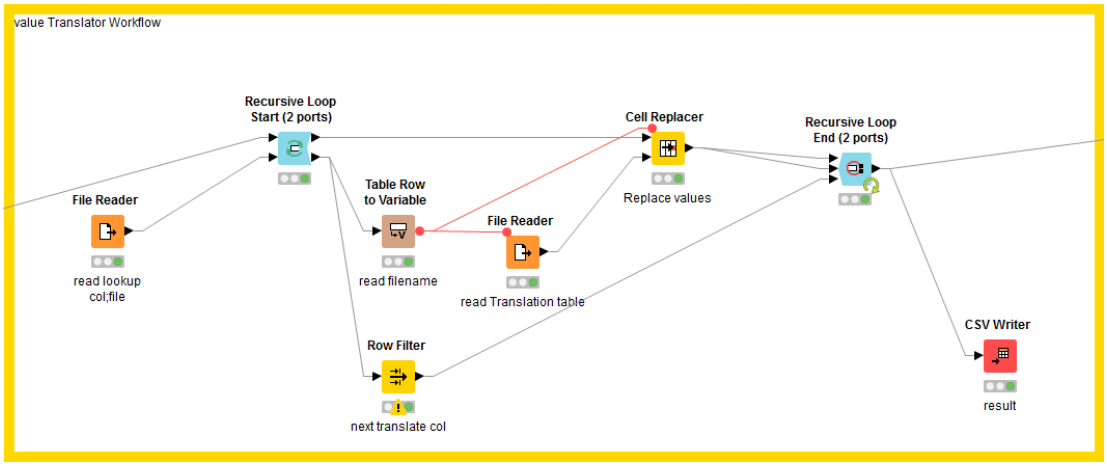


Abbildung 3.4.: Workflow: Bereinigung von Schreibfehlern und Abkürzungen

Die Notwendigkeit für eine Übersetzung von semantisch äquivalenten Werten war für die Merkmale *OBFSTS alpha*, *Werkstoff*, *Festigkeit*, *ABVOST alpha*, *GEWDUR alpha* und *AUSSCR alpha* gegeben. Daraus ergab sich eine von den entdeckten Werte-Äquivalenzen bereinigte Datentabelle.

3.3.2. Fehlende Daten

Im Anschluss an die Bereinigung der falschen Daten, wurde die Datentabelle auf fehlende Werte überprüft. Fehlende Werte sind in der zentralen Datentabelle vor allem auch durch die Datenintegration entstanden. Viele der integrierten Datentabellen enthielten durch die unterschiedlichen Tabellendefinitionen Merkmale, die in anderen Datentabellen nicht der Tabellendefinition angehörten. Dementsprechend besitzen diese Merkmale bei einigen Datensätzen keine Werte. Zudem sind fehlende Werte schlichtweg bei der Dateneingabe entstanden, in dem die entsprechenden Merkmale keinen Wert erhielten. Für die vorliegende Datenmenge wurden die zwei folgenden Methoden für den Umgang mit fehlenden Daten als sinnvoll angesehen:

1. Es besteht eine Relation zwischen Merkmalen
2. Globale Konstante

In Relation stehende Merkmale

Stehen Merkmale in Relation zueinander, so lässt sich aus einem dieser Merkmale der Wert des anderen ableiten bzw. vorhersagen. Eine Relation besteht beispielsweise zwischen den

Merkmalen *WerkstoffNr* und *Werkstoff*. Besitzt das Merkmal *WerkstoffNr* den Wert *1.1181*, kann daraus das Merkmal *Werkstoff* mit dem Wert *C35E* bestimmt werden, da sich hinter einer Werkstoff-Nummer immer derselbe Werkstoff verbirgt. Für eine Bereinigung der fehlenden Daten für in Relation stehende Merkmale, kann ähnlich wie bei der Übersetzung von falschen Daten in Unterabschnitt 3.3.1 eine Relationstabelle genutzt werden, deren Aufbau in Tabelle 3.4 dargestellt ist.

RelationFeature	**RelationFeatureValue**	**MissingValueFeature**	**PredictValue**
WerkstoffNr	1.1181	Werkstoff	C35E
WerkstoffNr	1.4301	Werkstoff	X5CrNi18-10
...	...	...	...

Tabelle 3.4.: Relationen von Merkmalen

Die Umsetzung der Bereinigung von Merkmalen zwischen denen eine Relation besteht ist in Abbildung 3.5 dargestellt. Über ein Java-Programm werden Regeln erstellt, die beim Einritt eines fehlenden Wertes in Kraft treten. Eine solche Regel baut sich aus den Werten der Relationstabelle auf und sieht wie folgt aus:

MISSING $MissingValueFeature$ AND $RelationFeature$ = “$RelationFeatureValue$“ => “$PredictValue$“

Diese Regel besagt, dass wenn der Wert eines Merkmals *MissingValueFeature* fehlt und das in Relation stehende Merkmal *RelationFeature* einen Wert *RelationFeatureValue* vorweist, dann wird der fehlende Wert durch den Wert *PredictValue* ersetzt. Die Ausführung dieser Regeln geschieht über den *Rule-Engine*-Node.

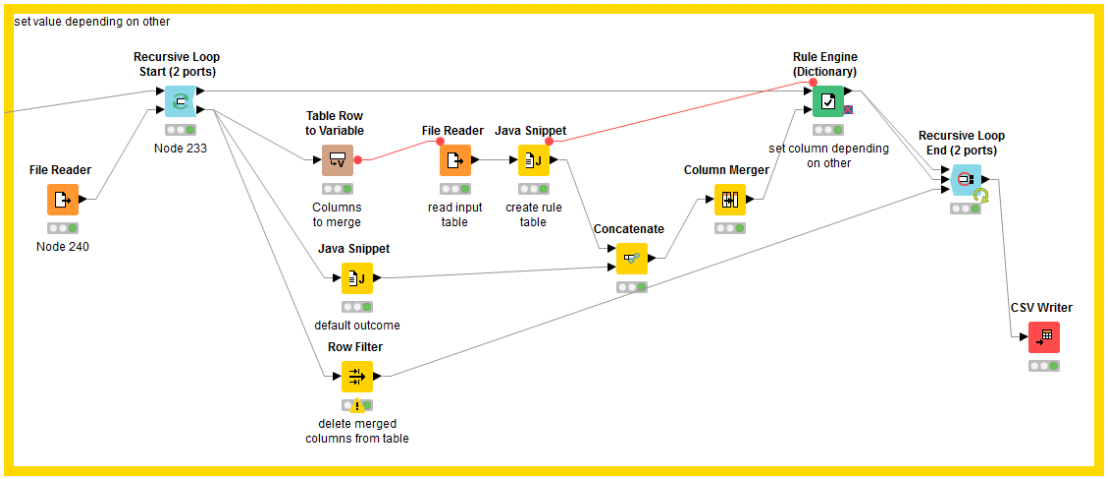


Abbildung 3.5.: Workflow: Setzen von Merkmalen mit Relationen

Globale Konstante

Durch die Bereinigung der in Relation stehenden Merkmale konnten natürlich nicht alle fehlenden Werte bereinigt werden, da eben nicht alle Merkmale eine Relation besitzen. Die nun offen gebliebenen fehlenden Werte werden durch eine globale Konstante ersetzt. Jedoch sollte bei der globalen Konstante noch unterschieden werden, ob ein Merkmal bereits einen Wert besitzt, der die gleiche Bedeutung eines fehlenden Wertes besitzt. Existiert solch ein Wert, so erhalten die fehlenden Werte über die globale Konstante denselben Wert, wie beispielsweise bei dem Merkmal *ABVOST alpha*, dass die Abnahmevorschrift einer Schraube beschreibt. Ist dieses Merkmal nicht gesetzt, so hat es die Bedeutung einer nicht vorhandenen Abnahmevorschrift. Da Merkmal *ABVOST alpha* bereits einen Merkmalswert *nein* besitzt, der die nicht Existenz einer Abnahmevorschrift beschreibt, sollte dieser verwendet werden. Wird ein abweichender Wert verwendet, bedarf das bereits gelöste Problem der syntaktisch unterschiedlichen Werte gleicher Semantik aus Unterabschnitt 3.3.1 einer erneuten Behandlung. Ist ein Wert mit gleicher Semantik eines fehlenden Wertes nicht vorhanden, so erhalten die Merkmale einen Standardwert, wie beispielsweise den Wert *-1* für numerische Merkmale und den Wert *NA* für nicht numerische Merkmale. Bei der Vergabe der Werte sollte darauf geachtet werden, dass Werte vergeben werden, die in den Merkmalen nicht auftreten können. In Abbildung 3.6 ist ein Workflow zum Setzen der fehlenden Werte dargestellt. Die individuell zu setzenden Werte, wie beim Merkmal *ABVOST alpha* mit *nein*, können in einer Tabelle, wie in Tabelle 3.5 dargestellt, definiert werden. Über diese Tabelle werden ähnlich, wie bei dem Workflow aus Abbildung 3.5, Regeln erstellt, die den definierten Wert anstelle des fehlenden Wertes setzen. Nach dem Setzen

der individuellen Werte werden über den *Missing-Value*-Knoten die bereits angesprochenen Standardwerte gesetzt. Die Standardwerte wurden für numerische Merkmale mit dem Wert *-1* und für alphanumerische Merkmale mit dem Wert *NA* versehen.

MissingValueFeature	Value
ABVOST alpha	nein
...	...

Tabelle 3.5.: Fehlende Werte durch individuelle Werte ersetzen

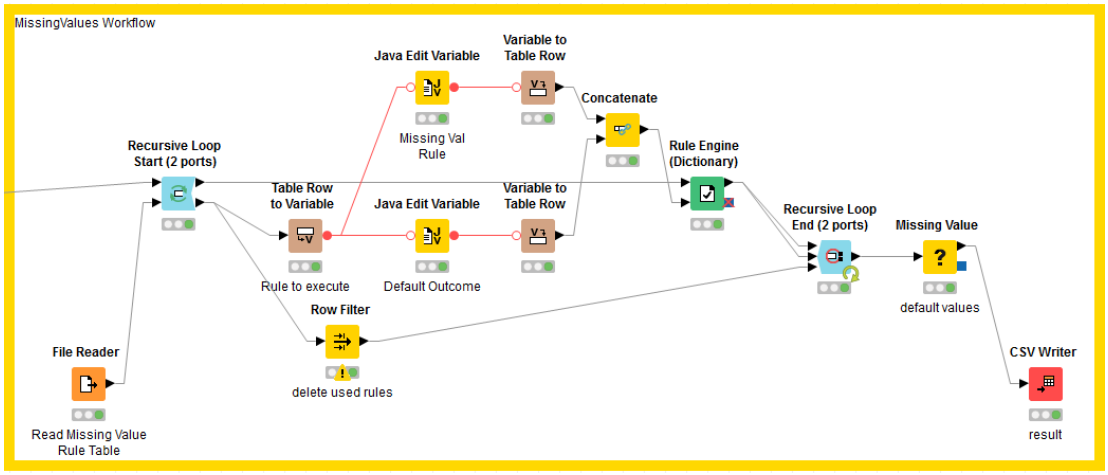


Abbildung 3.6.: Workflow: Bereinigung von fehlenden Werten

Nach Abschluss dieser Phase liegt eine von fehlenden Daten bereinigte Datentabelle vor.

3.4. Datentransformation

Wie bereits in Unterabschnitt 2.2.3 erwähnt, werden in der Datentransformation die Daten für das verwendete Data-Mining Verfahren vorbereitet. Um das verwendete Verfahren möglichst effizient auszuführen und um ein gutes Resultat zu erzielen wird zunächst eine Datenreduktion und anschließend eine Dimensionsreduktion vorgenommen. Die hier zur Anwendung kommenden Techniken für die Datenreduktion sind die Aggregation und die Datenkompression. Für die Attribut-Selektion wird neben dem Verfahren des Wrappers, eine zweite Attribut-Selektion mittels einem Verfahren über Filter angewandt. Da zwecks Aufgabenstellung eine Cluster-Analyse durchgeführt wird und die verwendeten Algorithmen numerische Werte benötigen,

müssen in dieser Phase die nominalen und ordinalen Daten zunächst durch numerische Werte kodiert werden. Anschließend werden diese Daten noch normalisiert.

3.4.1. Datenreduktion

Aggregation

Zunächst erfolgt eine Aggregation von Daten. Die Aggregation hat das Ziel mehrere Fakten zu einem Fakt zusammenzufassen. Die hier zur Anwendung kommende Aggregation erfolgt Spaltenweise, das heißt es werden mehrere Merkmale zu einem zusammengefasst. Die Entscheidung Merkmale zu aggregieren erfolgt auf Basis des Anwendungswissens. Die Merkmale *GEWEN1 alpha* und *GEWEN2 alpha* beschreiben ein Gewindeende 1, sowie ein Gewindeende 2 und treten nur in Kombination bei bestimmten Schraubentypen auf. Aufgrund dessen werden die Merkmale zu einem Merkmal aggregiert. Für die Durchführung von Aggregationen wurde der in Abbildung 3.7 abgebildete Workflow erstellt. Dieser Workflow benötigt als Grundlage eine Aggregationstabelle, die den Aufbau der Tabelle 3.6 besitzt. Die Tabelle enthält die Merkmale *Feature1* und *Feature2*, die aggregiert werden sollen, sowie einen Namen *NewFeatureName* für das aggregierte Merkmal. Die beiden Merkmale werden dabei durch ein definiertes Zeichen getrennt.

Feature1	**Feature2**	**NewFeatureName**
GEWEN1 alpha	GEWEN2 alpha	GEWENDEN
...	...	...

Tabelle 3.6.: Merkmals-Aggregationen

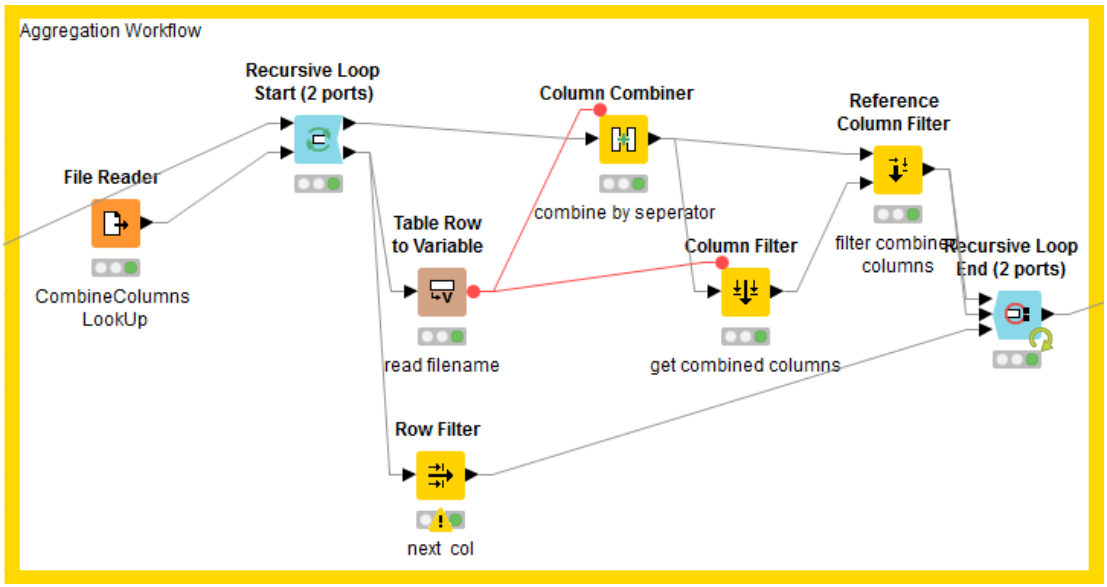


Abbildung 3.7.: Workflow: Merkmale aggregieren

Die Aggregation wird nur bei den bereits erwähnten Merkmalen *GEWEN1 alpha* und *GEWEN2 alpha* durchgeführt. Die Durchführung sollte vor Kodierung und Normalisierung geschehen, da ansonsten aus numerischen Attributen wegen des Trennzeichens alphanumerische Attribute entstehen.

Datenkompression

Nachdem die Aggregation ausführt wurde und Merkmale zusammengefasst hat, kommt die Datenkompression zum Einsatz. Bei der Datenkompressionen steht die Verringerung der Merkmalswerte bzw. das Zusammenfassen mehrerer Werte zu einem im Vordergrund. Da das Attribut *Norm alpha* sehr viele Werte enthält, soll dies als Beispiel für die Durchführung der Datenkompression dienen. Es gibt eine Menge von Normen für Schrauben hinter denen sich neben technischer Artefakte auch der Schraubentyp verbirgt. Für eine Verringerung der Werte können diese in *Schraubentyp Norm* zusammengefasst werden. Das heißt es werden etwa die Normen des Schraubentyps Sechskantschraube *DIN 561, DIN 564, ..., ISO 4018* zu einem Attribut *Sechskantschrauben Norm* zusammengefasst. Die Umsetzung der Datenkompression ist im Workflow in Abbildung 3.8 dargestellt.

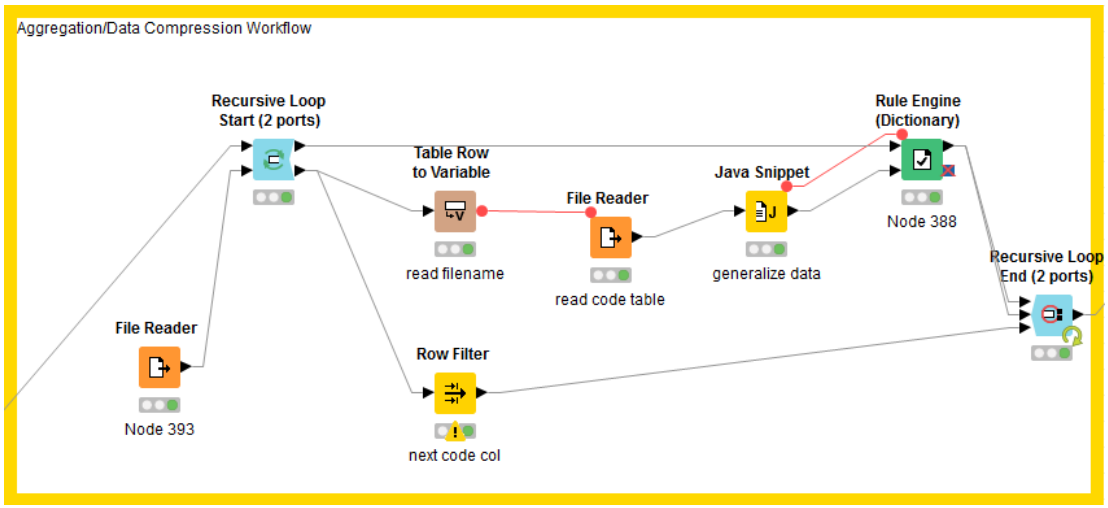


Abbildung 3.8.: Workflow: Datenkompression

Auf Basis einer Kompressionstabelle erstellt der Workflow, analog zu den bisherigen Workflows, Regeln zur Übersetzung bzw. Komprimierung. Die Kompressionstabelle hat den folgenden Aufbau:

FeatureValue	**CompressedValue**
DIN 561	Sechskantschrauben Norm
DIN 564	Sechskantschrauben Norm
...	...

Tabelle 3.7.: Datenkompressionen

Der Workflow übersetzt also die Werte *FeatureValue* auf einen komprimierten Wert *CompressedValue*. Zur Anwendung kommt die Datenkompression bei den Merkmalen *Norm alpha*, *GEWDUR alpha*, *SICHNA alpha* und *ABVOST alpha*. Auch die Datenkompression sollte vor der Kodierung, sowie der Normalisierung durchgeführt werden, denn die kodierten Werte würden eine Kompression erschweren, da zunächst der ursprüngliche Wert des kodierten Wertes ermittelt werden müsste.

3.4.2. Dimensionsreduktion

Nachdem nun die Datenreduktion durchgeführt wurde, sollen nun die für die Analyse relevanten Merkmale identifiziert werden. Wie anfangs erwähnt werden dazu zwei verschiedene Verfahren eingesetzt. Zunächst wird das Verfahren der Attribut-Selektion über Filter beschrieben, ehe sich dem Verfahren des Wrappers gewidmet wird.

Filter

Missing-Value-Filter

Zunächst kommt ein *Missing-Value-Filter* zu Anwendung. Dieser Filter entfernt Merkmale, die zu einem gewissen Grenzwert fehlende Werte vorweisen. Ist der Anteil der fehlenden Werte sehr hoch, so ist es unwahrscheinlich, dass der Informationsgehalt dieses Merkmals bedeutend ist. Bei der Anwendung ist darauf zu achten, dass der Filter vor dem Setzen von fehlenden Werten durch eine globale Konstante, jedoch nach dem Setzen der fehlenden Daten, die in Relation zu einem anderen Merkmal stehen, ausgeführt wird, da dieser ansonsten keine Wirkung hat. In KNIME ist bereits ein *Missing-Value-Filter* mitgeliefert. Dieser wird über einen Grenzwert des prozentualen Anteils fehlender Werte konfiguriert. Der Filter wurde auf einen Grenzwert von 97,5% fehlender Werte konfiguriert. Dadurch wurden die folgenden Merkmale für die Daten-Analyse als irrelevant bewertet:

Feature
TEAB01 alpha
TEAB02 alpha
ANSKUP alpha
LIEUMF alpha
KOPFBM numer.
PRODKL alpha
DUSCHF numer.
HERSTE alpha
SCHABR numer.
VERPEH alpha
SCHLFO alpha
FRM962 alpha
AEENDE alpha
ANMUTE numer.
TELSCH alpha
TECLIF alpha
DUSCHB alpha
KKLASS alpha
AUSSCR alpha

Tabelle 3.8.: Merkmale mit fehlenden Werten >= 97,5%

Die gefilterte Datentabelle wurde als Grundlage für die weitere Verarbeitung sowohl für die der Filter als auch für die des Wrappers genutzt.

High-Correlation-Filter

Ein weiterer Filter, der für die Attribut-Selektion zur Anwendung kommt, ist der High-Correlation-Filter. Die Aufgabe dieses Filters besteht darin, redundante Attribute aus der Datentabelle zu filtern. Zur Identifizierung der redundanten Attribute kommt der Korrelationskoeffizient von Pearson zum Einsatz. Dieser Koeffizient stellt ein Maß für den Grad des linearen Zusammenhanges zwischen Merkmalen und kann Werte zwischen -1 und 1 annehmen. Beträgt der Wert 1 bzw. -1 besteht ein vollständiger linearer Zusammenhang zwischen den betrachteten Merkmalen, wohingegen bei einem Koeffizienten gegen 0 kein Zusammenhang zwischen den Merkmalen besteht. Da beispielsweise die Attribute *WerkstoffNr* und *Werkstoff* in Relation stehen und auf Basis dieser die fehlenden Werte in Abschnitt 3.3.2 gesetzt wurden, ist es durchaus wahrscheinlich, dass in diesen Attributen Redundanzen erzeugt wurden. Der High-Correlation-Filter wurde auf einen Grenzwert von 0.9 konfiguriert, wodurch Merkmale mit einem Koeffizienten über diesem Grenzwert entfernt werden. Nach Ausführung des High-Correlation-Filters wird, wie bereits vermutet, auch das Merkmal *Werkstoff* gefiltert. Die gefilterten Merkmale sind:

Feature
GEWLG2 numer.
Werkstoff
GEWENDEN

Tabelle 3.9.: Merkmale mit einer Korrelation >= 0.9

Low-Variance-Fiter

Nachdem der High-Correlation-Filter bereits einige Attribute gefiltert hat, kommt nun der Low-Variance-Fiter zur Anwendung. Dieser Filter ermittelt über das Maß der Varianz den Informationsgehalt eines Merkmals. Wurde eine Varianz eines Merkmals von 0 ermittelt, so ist dieses Merkmal nicht relevant, da es einen konstanten Wert enthält. Weist die Varianz eines Merkmals einen Wert nahe 0 auf, so ist der Informationsgehalt dieses Merkmals gering. Der Low-Variance-Filter wurde auf einen Grenzwert von 0.01 konfiguriert, wodurch Merkmale mit einer Varianz unterhalb dieses Grenzwertes gefiltert werden. Durch die Ausführung des Low-Variance-Filters werden die folgenden Merkmale gefiltert:

Feature
GEWDUR alpha
LANGE numer.
GEWLGE numer.
FORM alpha

Tabelle 3.10.: Merkmale mit einer Varianz <= 0.01

Die Umsetzung der Korrelation und Varianz Filter ist in Abbildung 3.9 zu sehen. Zu beachten ist, dass diese Filter nur auf numerische, sowie normalisierte Werte angewandt werden können. Die Kodierung wird später im Unterabschnitt 3.4.3 und die Normalisierung Unterabschnitt 3.4.4 thematisiert.

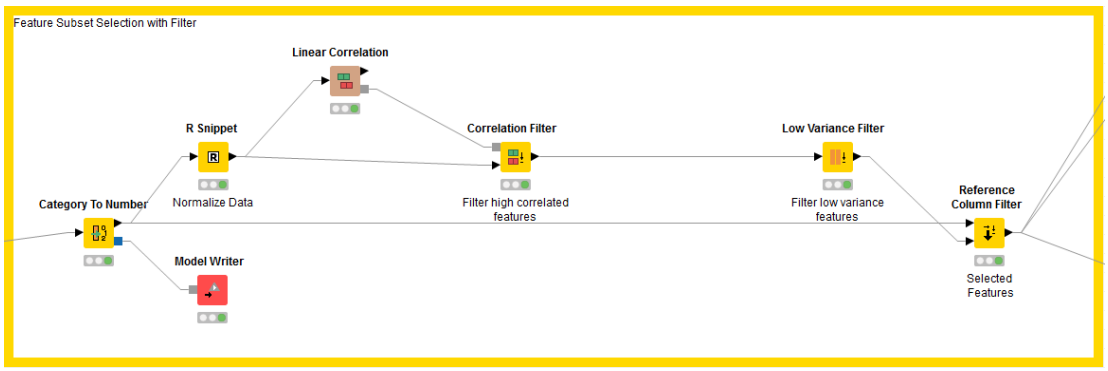


Abbildung 3.9.: Workflow: Dimensionsreduktion mit Filtern

Nach Ausführung der Filter bleiben 10 Merkmale erhalten. Das Merkmal *Teilebezeichnung* ist in der Menge der nicht gefilterten Merkmale enthalten, wird aber auf Basis von Hintergrundwissen ebenfalls gefiltert. Dieses Merkmal ist ein Aggregat aus den Merkmalen *GEWDUR alpha*, *LANGE numer.* und dem Schraubentyp, der sich hinter dem Merkmal *NORM alpha* verbirgt. Zudem wird das Identifizierungsmerkmal *Teile-Nr* gefiltert, da dieses Merkmal aus einzigartigen Einträgen besteht und somit keinen Beitrag zur Findung von Gruppierungen leistet. Somit sieht die Merkmalsauswahl über die Filter wie folgt aus:

SCHLUW numer.	KOPFHH numer.	ABVOST alpha	SICHNA alpha
KOPFBR numer.	GEWLG1 numer.	OBFSTS alpha	NORM alpha
Festigkeit	WerkstoffNr		

Wrapper

Die zweite Klasse der Algorithmen für die Attribut-Selektion, die hier zur Anwendung kommt, ist der Wrapper. Wie bereits in Unterabschnitt 2.2.3 erwähnt führt ein Wrapper für alle Unterräume des Merkmalsraumes ein Clustering durch und bewertet dieses anhand eines ausgewählten Kriteriums. Für die Durchführung der Cluster-Analyse kommt der *k-Means*-Algorithmus zum Einsatz. Als Bewertungskriterium wurde der *Silhouetten-Koeffizient* gewählt, da dieser für die Verwendung einer variablen Clusteranzahl gut geeignet ist. Da es sich bei dem *k-Means*-Algorithmus um ein Abstand-basiertes Verfahren handelt, ist auch beim Wrapper zunächst eine Kodierung und Normalisierung nötig. Da das mehrmalige Durchführen einer Cluster-Analyse mit jeder möglichen Merkmalsmenge eine hohe Laufzeit zur Folge hat, wurde sich nach einer Alternative umgeschaut. Bei diesem Verfahren wird für jedes Merkmal individuell ein Clustering durchgeführt und bewertet. Dabei wird der anfangs leeren Merkmalsuntermenge iterativ das am besten bewertete Merkmal hinzugefügt. Für die in jedem Durchlauf neu ermittelte Merkmalsuntermenge wird ebenfalls mit jeder Iteration ein Clustering und eine Bewertung durchgeführt. Dieses Vorgehen ist im Pseudo-Code 3.2 umgesetzt.

Algorithm 3.2 Pseudocode: Wrapper

```
1:  F ← {AllFeatures}
2:  fss ← {∅}
3:  for all f ∈ F do
4:      c ← Clustering(f)
5:      eval[f] ← Evaluate(c)
6:  end for
7:  for i < length(eval) do
8:      bestFeature ← getBestFeature(eval)
9:      fss ← fss ∪ bestFeature
10:     fssC[i] ← Clustering(fss)
11: end for
```

Nach der Ausführung des Wrappers kann über das Bewertungsmaß des Silhouetten Koeffizienten beurteilt werden, welcher der entstandenen Merkmalsunterräume ein gutes Ergebnis liefert. In Abbildung 3.10 ist das Ergebnis eines Durchlaufes dargestellt. Darin ist zu sehen, dass mit 4 Merkmalen ein Koeffizient nahe 1 erreicht wird. Dabei sollte zunächst ein Merkmalsunterraum gewählt werden, der mit möglichst vielen Merkmalen ein gutes Ergebnis erreicht, um möglichst wenig Informationen zu verlieren. Die Auswahl fiel daher auf den Merkmalsraum mit 11 Merkmalen, da ab 12 Merkmalen eine stetige Verschlechterung des Resultats eintrat.

Ebenfalls denkbar wäre eine Variante, in der die besten *n* Merkmale ausgewählt werden und der Algorithmus terminiert, sobald die Anzahl *n* erreicht ist.

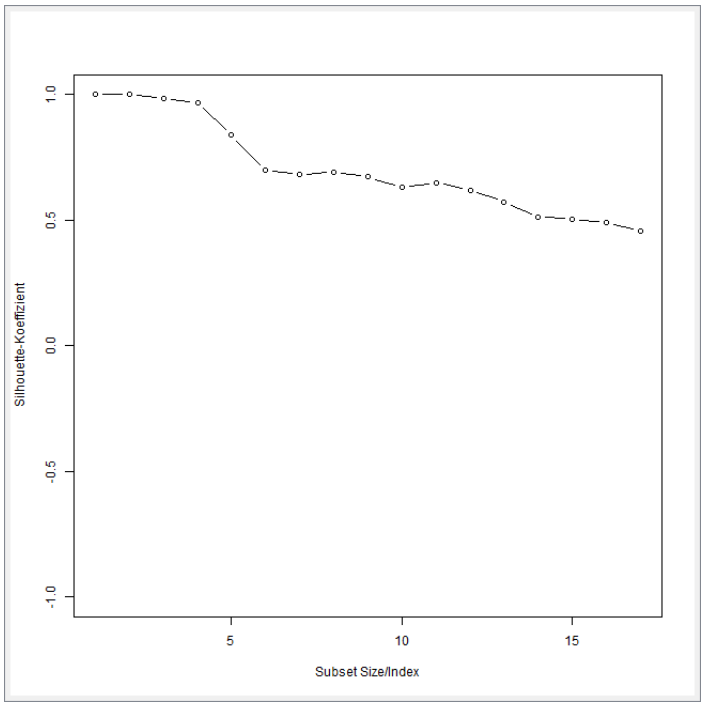


Abbildung 3.10.: Wrapper: Silhouetten Koeffizienten der ermittelten Merkmalsmengen

Die 11 Merkmale, die durch den Wrapper ermittelt wurden, sind die folgenden:

GEWDUR alpha	ABVOST alpha	OBFSTS alpha	SICHNA alpha
NORM alpha	Festigkeit	GEWLG1 numer.	GEWENDEN
FORM alpha	WerkstoffNr	Werkstoff	

Im Gegensatz zu den Filtern, erkennt der Wrapper keine hoch korrelierenden Merkmale wie *Werkstoff* und *WerkstoffNr*, da dieser die Merkmale im Einzelnen betrachtet. Dafür werden mit gewählter Untermenge die Merkmale *Teilebezeichnung* und *Teile-Nr* als nicht relevant angesehen. Bei einem Vergleich der ermittelten Merkmalsmengen der Filter, sowie des Wrappers ist zu erkennen, dass mit unterschiedlichen Verfahren durchaus unterschiedliche Merkmale selektiert werden.

3.4.3. Kodierung

Da die zur Anwendung kommendem Cluster-Analyse-Verfahren numerische Werte erwarten, sind die nicht numerischen Werte zunächst in numerische zu kodieren. Dafür ist zunächst

eine Überprüfung der Datentypen nötig. Handelt es dabei um metrische Attribute, ist keine Kodierung nötig. Liegt ein Merkmal mit ordinalen Daten wie das Merkmal *GEWENDEN* vor, müssen diese Daten gemäß ihrer Ordnung kodiert werden. Bei dem vorliegenden Anwendungsfall der Schrauben ist eine Bewertung der Merkmale, ob ein Wert *besser* oder *schlechter* als ein anderer ist, stark vom Verwendungszweck der Schraube abhängig. Wird beispielsweise eine Schraube im Innenbereich verwendet, stehen eventuell optische Anforderungen im Vordergrund, wohingegen im Außenbereich ein guter Korrosionsschutz von Bedeutung ist. Aus diesem Grund liegen zum Großteil nominale, sowie metrische Daten vor. Einzig das bereits erwähnte Merkmal *GEWENDEN* enthält ordinale Daten, da sich die Größe der Gewindeenden dahinter verbirgt. Die nominalen Daten werden durch Integer-Werte kodiert werden, sodass eine Kodierung für das Merkmal *OBFSTS alpha* wie folgt aussehen könnte:

$$
\begin{aligned}
NA &\rightarrow 0 \\
blank &\rightarrow 1 \\
feuerverzinkt &\rightarrow 2 \\
&\dots \\
galv.verz. &\rightarrow n
\end{aligned}
$$

Eine individuelle Kodierung, wie sie bei ordinalen Daten von Nöten ist, kann über den Workflow in Abbildung 3.11 vorgenommen werden. Der Workflow arbeitet auf Grundlage einer Kodierungstabelle. Die Funktionsweise dieses Workflows ist ähnlich der, des in Abbildung 3.4 dargestellten Workflows, denn letztendlich findet hier auch eine Werte Übersetzung statt. Für eine Kodierung der nominalen Daten kann der *Category-to-Number*-Knoten genutzt werden. Dieser kodiert die kategorischen Werte mit einem wählbaren Inkrement auf einen numerischen Wert.

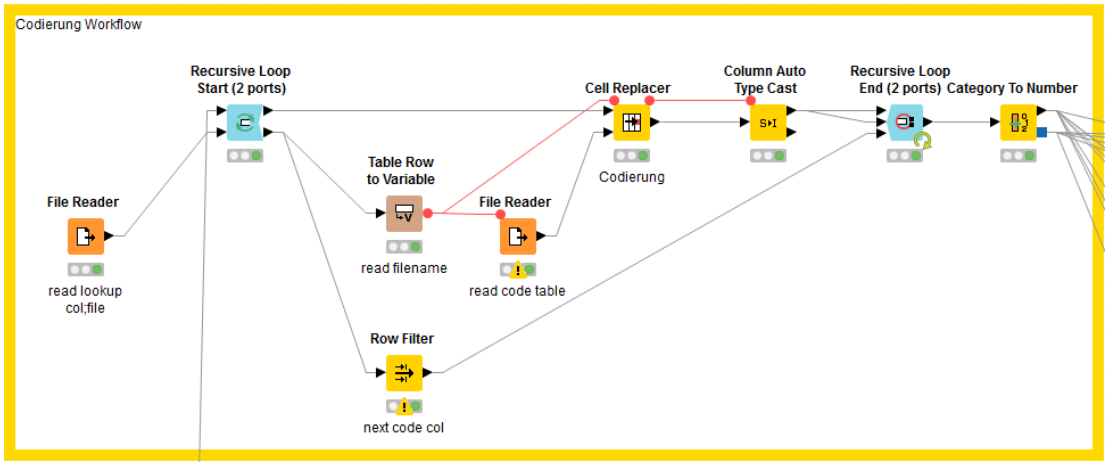


Abbildung 3.11.: Worfklow: Individuelle Kodierung

Nach Ausführung des Workflows ist die Datentabelle, die ausschließlich aus numerischen Werten besteht, für die Cluster-Analyse-Verfahren vorbereitet.

3.4.4. Normalisierung

Da beispielsweise das Merkmale *LANGE numer.* im Vergleich zu anderen Merkmalen sehr hohe Werte aufweist, könnte dieses Merkmal einen großen Einfluss auf das Resultat nehmen. Um den hohen Einfluss der Merkmale zu vermeiden, wird eine Normalisierung der Werte vorgenommen. Umgesetzt wurde die Normalisierung in einem kleinen R Programm, welches über einen *R-Snippet-Node* in KNIME eingebettet ist. Der Pseudo-Code 3.3 stellt das prinzipielle Vorgehen der Normalisierung dar.

Über eine Tabelle kann eine Gewichtung der Merkmale eingelesen werden. Ist ein Merkmal in der Tabelle nicht enthalten, so wird dieses Merkmal auf das Standardintervall [0,1] normalisiert. Zunächst wurden alle Merkmale auf das Standardintervall [0,1] normiert, um einen nicht gewollten, großen Einfluss eines Merkmals auf das Resultat zu verhindern. Bei nicht zufriedenstellenden Ergebnissen können die Intervalle der Merkmale, wie in Unterabschnitt 2.2.3 bereits erwähnt, gemäß ihrer geschätzten Bedeutung auf ein einflussreicheres Intervall angepasst werden.

3.5. Data Mining

Nachdem die Daten für die Cluster-Analyse vorbereitet wurden, gilt es diese durchzuführen. Die Wahl der Clustering-Algorithmen fiel auf ein partitionierendes und ein dichtebasiertes

Algorithm 3.3 Pseudocode: Normalisierung

```
function NormalizeFeature(x, w)
    x ← w * (x − min(x))/max(x) − min(x))
    return x
end function

function NormalizeAll(dataTable)
    weightTable < −readTable(path)
    for all columns ∈ dataTable do
        if columns.name ∈ weightTable then
            w ← getValueFromWeightTable(columns.name)
            columns ← NormalizeFeature(columns, w)
        else
            columns ← NormalizeFeature(columns, 1)
        end if
    end for
    return dataframe
end function
```

Verfahren. Bei dem partitionierenden Verfahren wurde sich für den *k-Means*-Algorithmus entschieden, da dieser in der Literatur oftmals als Referenzverfahren genannt wird. Für ein dichtebasiertes Verfahren als zweites Clustering-Verfahren wurde sich entschieden, da diese Verfahren eine gute Ausreißer-Erkennung haben und ebenfalls konträr zu den partitionierenden Verfahren konvexe Cluster bilden können. Als speziellen Algorithmus dieser Verfahren wurde der DBScan-Algorithmus gewählt. Im Folgenden soll kurz die Arbeitsweise der Algorithmen beschrieben werden. Umgesetzt werden die Cluster-Algorithmen in der Programmiersprache R, die über einen *R-Snippet*-Knoten in KNIME eingebunden werden. KNIME selbst bietet zwar auch Clustering-Verfahren an, jedoch ist die Durchführung in R etwas komfortabler, zumal R bereits einige Techniken für die Ermittlung der Parameter bereitstellt. Die nach Ausführung der Algorithmen vorliegenden Daten bilden die Grundlage für die Bewertung durch einen Experten in der Folgephase.

3.5.1. k-Means

Der *k-means*-Algorithmus von Macqueen (1967) ist, wie bereits erwähnt, ein sehr populäres Verfahren. Dieses Verfahren gehört zur Klasse der Mittelpunkt-Verfahren und nutzt dementsprechend Centroide zur Repräsentation der Cluster. Bei der Clusterbildung mittels *k-means* werden zunächst *k* zufällig gewählte Clusterzentren bzw. Centroide gebildet. Nun werden mit

jeder Iteration die Objekte den Clustern zugeordnet, zu denen sie den geringsten Abstand haben. Der Abstand eines Objektes zu einem Cluster wird über das Abstandsmaß der euklidischen Distanz ermittelt. Im Anschluss werden die Centroide durch den Mittelwert, der ihn zugeordneten Objekte neu berechnet. Wird kein Objekt einem anderen Cluster, als dem es ohnehin schon angehört, zugeordnet, terminiert der Algorithmus.
Die Schnittstelle des verwendeten *k-Means*-Algorithmus aus der R-Standardbibliothek sieht wie folgt aus:

$$kmeans(data, k, algorithm = "MacQueen", nstart = 25)$$

Der Parameter *data* steht für die zu analysierenden Daten und das *k* für die Anzahl der zu bildenden Cluster. Da die Wahl der Anfangspartitionen wie erwähnt zufällig geschieht, sorgt der Parameter *nstart* dafür, dass dies 25 mal durchgeführt wird. Aus diesen 25 Versuchen wird die beste Anfangspartition gewählt. Über den Parameter *algortihm* sind auch andere *k-Means*-Algorithmen wählbar. Da die Anzahl der zu bildenden Cluster im Vorhinein unbekannt ist, gestaltet sich die Wahl für diesen Parameter schwierig. Eine Variante wäre mit verschiedenen *k* zu experimentieren und zu ermitteln mit welcher Clusteranzahl *k* das beste Ergebnis entstanden ist. Da die Bewertung der Resultate für verschiedene *k* durch einen Experten sehr aufwendig wäre, wurde nach einer automatisierten Möglichkeit gesucht. Um die Bewertung automatisiert durchzuführen, wird ein Bewertungsmaß, wie der Silhouetten-Koeffizient, benötigt, der die Qualität von Clustern bewertet. Um ein optimales *k* zu ermitteln, wird für jedes *k* im Intervall [2:kmax] ein Clustering durchgeführt und mittels dem Silhouetten-Koeffizienten bewertet. Das Clustering-Resultat mit dem besten Koeffizienten liefert, im besagten Intervall, die beste Anzahl von Clustern. Dieses Vorgehen ist im Pseudo-Code im Algorithmus 3.4 dargestellt. Die Berechnung des Silhouetten-Koeffizienten wird über das Paket *'cluster'* bereitgestellt.

Algorithm 3.4 Pseudocode: Ermittlung einer Clusteranzahl

```
k ← 2
for k ≤ kmax do
    cluster ← kmeans(data, numOfCluster = k)
    sil[k] ← Silhouette(cluster, dist(data))
end for
return indexOf(max(sil))
```

Nach Ermittlung des Parameters *k* und der Ausführung des Clusterings, sind die Daten mit dem zugeordneten Cluster in Form einer Zahl gelabelt.

3.5.2. DBScan

Nachdem die Clusterbildung mit dem Referenzverfahren *k-means* durchgeführt wurde, folgt nun mit dem *DBScan*-Algorithmus das zweite Verfahren. Da es sich um ein dichtebasiertes Verfahren handelt, werden zunächst die Parameter zur Definition der Dichte dargestellt und die Arbeitsweise dieses Verfahrens beschrieben. Die Dichte ist wie folgt definiert:

- Der Radius $\epsilon > 0$.
- *minPoints* gibt die Mindestanzahl der Nachbarschaftsobjekte vor.

Bei dem *DBScan*-Algorithmus werden zunächst alle Objekte als "unbearbeitet" gekennzeichnet. Aus der Datenmenge wird ein zufälliges Objekt gewählt, zu dem die in der unmittelbaren ϵ-Umgebung liegenden Objekte ermittelt werden. Ist die Anzahl der ermittelten Objekte *minPoints*, die in der Nachbarschaft liegen groß genug, so bilden diese ein Cluster. Ist die Anzahl nicht ausreichend, werden die Objekte als "Rauschen" markiert. Wenn ein neues Cluster entsteht, wird geprüft, ob die neuen Objekte weitere *minPoints* Objekte in ihrer ϵ-Umgebung bzw. ihrer Nachbarschaft haben. Ist das der Fall, erweitern diese Objekte das bereits vorhandene Cluster. Besitzt keines der Objekte mehr den Status "unbearbeitet", terminiert der Algorithmus. Der *DBScan*-Algorithmus benötigt im Gegensatz zu dem *k-Means*-Algorithmus keine Angabe der zu bildenden Cluster. Jedoch besteht bei diesem Algorithmus das Problem einer geeigneten Wahl für die Parameter ϵ und *minPoints*. Für die Wahl des Parameters ϵ kann ein k-nearest-neighbour-Histogramm erstellt werden. Dafür werden für jedes Objekt die Abstände der k-nächsten Nachbarn ermittelt. Die ermittelten Abstände werden dann in aufsteigender Reihenfolge in einem Diagramm dargestellt. Für die Ermittlung der k-nächsten Nachbarn ist die Angabe eines *k* nötig, die dem Wert *minPoints* entsprechen sollte. In dem erstellten Diagramm sollte ein scharfer Übergang zu sehen sein, der eine gute Wahl für ϵ darstellt. In Abbildung 3.12 ist ein solches Diagramm zu sehen. Dort ist ein Richtwert von ungefähr 0.09 für ϵ zu sehen. Für eine gute Wahl des Parameters *minPoints* sollte mit verschiedenen Werten experimentiert werden.

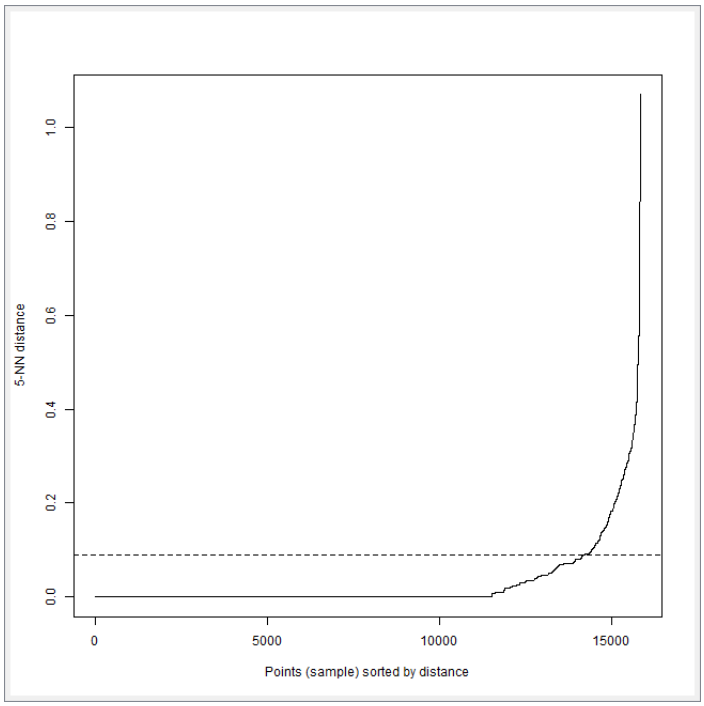


Abbildung 3.12.: Ermittlung von Epsilon für DBScan

Den Plot des *k-nearest-neighbour*-Histogramms stellt das Paket *'dbscan'* über die Funktion

$$kNNdistplot(data, k)$$

zur Verfügung. Über dasselbe Paket wird auch der *DBScan*-Algorithmus zu Verfügung gestellt, dessen Schnittstelle wie folgt aussieht:

$$dbscan(data, eps = 0.09, MinPts = 30)$$

Der Parameter *data* steht für die zu analysierenden Daten, *eps* und *MinPts* für die eben beschriebenen Parameter ϵ und *minPoints* zur Definition der Dichte. Die Parameter wurden für die Merkmalsuntermenge resultierend aus der Attribut-Selektion über Filter mit $\epsilon = 0.09$ und $minPoints = 30$ gewählt. Für die Merkmalsmenge, die durch den Wrapper ermittelt wurde, fiel die Wahl auf die Parameter $minPoints = 30$, sowie $\epsilon = 0.04$. Nach Ausführung des Clusterings liegen die Daten ebenfalls mit einem Cluster gelabelt vor. Als Besonderheit zu erwähnen ist, dass das Cluster 0 jene Objekte repräsentiert, die durch den Algorithmus als "Rauschen" markiert wurden.

4. Evaluation

In diesem Kapitel geht es um die Evaluation, der letzten Phase des KDD Prozesses. In dieser Phase werden die Ergebnisse aus Kapitel 3 im Hinblick auf die Kriterien *Verständlichkeit*, *Neuartigkeit* und *Nützlichkeit* aus Unterabschnitt 2.2.5 überprüft. Des Weiteren werden bei Nichterfüllen der Kriterien geeignete Maßnahmen, die eine Verbesserung des Resultats zur Folge haben sollen, eingeleitet. Im Folgenden dieses Kapitels werden die Resultate und deren Bewertung durch einen Experten beschrieben und an Beispielen veranschaulicht. Dabei werden die Resultate nach dem Verfahren der Merkmalsauswahl sowie dem verwendeten Clustering-Algorithmus untergliedert. Zu Beginn eines Abschnittes werden die durchgeführten Maßnahmen zur Erzielung eines verwertbaren Resultats beschrieben, ehe die konkreten Einstellungen, die dafür nötig waren, dargestellt werden. Darauffolgend wird das Resultat charakterisiert und mittels dem Experten erneut evaluiert. Zum Abschluss dieses Kapitels werden sowohl die Auswahlverfahren als auch die Clustering-Verfahren gegenübergestellt und hinsichtlich ihrer Nutzbarkeit in diesem Kontext bewertet.

4.1. Durchlauf 1

4.1.1. Merkmalsauswahl über Filter

Clusterbildung mit k-Means

Mit dem *k-Means*-Algorithmus und der Merkmalsauswahl über die Filter wurde eine Clusteranzahl k von 20 ermittelt. Der Silhouetten Koeffizient dieser Clusterbildung beträgt 0.465938, woraus der Fund einer schwachen Struktur hervorgeht. Die Bewertung des Experten wird im Folgenden exemplarisch dargestellt. Das Cluster 1 wird als eine verständliche und sinnvolle Gruppierung bewertet. Dieses Cluster enthält ausschließlich den Schraubentyp der Stiftschraube. Jedoch sind Schrauben des gleichen Schraubentyps auch in anderen Clustern mit weiteren Schraubentypen vermischt. An den in Tabelle 4.1 exemplarisch dargestellten Datensätzen soll erläutert werden, warum diese Clusterbildung für den Experten unverständlich ist. Bei einem Vergleich der Datensätze 1 und 3, die dem gleichen Cluster zugeordnet wurden, fällt auf, dass diese sich in den Merkmalen *ABOST alpha*, *GEWLG1 numer.*, *OBFSTS alpha*, *NORM*

alpha und *WerkstoffNr* unterscheiden. Dem hingegen unterscheiden sich der Datensatz 2 und der Datensatz 1 nur in den Merkmalen *Festigkeit* und *WerkstoffNr*, wurden jedoch unterschiedlichen Clustern zugeordnet. Aufgrund der Tatsache, dass die Datensätze 1 und 3 trotz der augenscheinlich größeren Unterschiede dem gleichen Cluster zugeordnet wurden, bewertet der Experte die Clusterbildung als nicht verständlich.

	Datensatz 1	Datensatz 2	Datensatz 3
SCHLUW numer.	-1,0	-1,0	-1,0
KOPFHH numer.	NA	NA	NA
ABVOST alpha	gestempelt	gestempelt	nein
SICHNA alpha	NA	NA	NA
KOPFBR numer.	-1,0	-1,0	-1,0
GEWLG1 numer.	15,0	15,0	-1,0
OBFSTS alpha	galv.verz.	galv.verz.	feuerverzinkt
NORM alpha	Stiftschrauben Norm	Stiftschrauben Norm	Spannschrauben Norm
Festigkeit	NA	5.6	NA
WerkstoffNr	1.1181	NA	0.0004
Cluster	10	1	10

Tabelle 4.1.: Evaluation: k-Means mit Merkmalsauswahl über Filter

Dieses beispielhaft dargestellte Verhalten spiegelt sich im ganzen Resultat wider und betrifft nicht nur die dargestellten Cluster. Eben aufgrund der beschriebenen unverständlichen Clusterbildung und einiger Ausreißer, die das Resultat zusätzlich trüben, bewertete der Experte dieses Resultat als unbrauchbar.

Clusterbildung mit DBScan

Der *DBScan*-Algorithmus wurde anfangs mit einem $\epsilon = 0.09$ und *minPoints* = 30 konfiguriert. Das Resultat sind 17 Cluster und ein Silhouetten Koeffizient von 0.1275339. Der Koeffizient besagt, dass keine wesentliche Struktur gefunden wurde. Jedoch spiegelt sich die Bewertung des Silhouetten Koeffizienten nicht in der Bewertung des Experten wider, denn die entstandenen Cluster liefern sehr wohl verständliche Gruppierungen. In der Tabelle 4.2 sind 4 exemplarische Cluster dargestellt, anhand derer die entstandenen Gruppierungen erläutert werden sollen. Zunächst wirkt es so, dass das Merkmal *NORM alpha* einen dominierenden Einfluss auf das Resultat hat, denn alle Cluster bestehen aus einem einzigen Typ von Schrauben. Zudem fand

anscheinend aufgrund der Merkmale *ABVOST alpha*, *SICHNA alpha*, *OBFSTS alpha*, *Festigkeit* und *WerkstoffNr* eine weitere Untergliederung statt. Erkennbar ist das beispielhaft an dem Cluster 4 und 15, die sich in den genannten Merkmale nur im Merkmal *Festigkeit* unterscheiden. Bei den Clustern 4 und 6 hingegen liegt der Unterschied im Merkmal *SICHNA alpha*.

Cluster	2	4	15	6	8
SCHLUW numer.	8 - 46	7 - 55	13 - 19	10 - 55	8 - 46
KOPFHH numer.	3,5 - 18,7	2,8 - 18,7	9 - 11	4 - 20,5	3,5 - 18,7
ABVOST alpha	nein	gestempelt	gestempelt	gestempelt	gestempelt
SICHNA alpha	NA	NA	NA	AD2000-W7/TRD106	NA
KOPFBR numer.	-1,0	-1,0	-1,0	-1,0	-1,0
GEWLG1 numer.	-1,0	-1,0	-1,0	-1,0	-1,0
OBFSTS alpha	galv.verz.	galv.verz.	galv.verz.	galv.verz.	blank
NORM alpha	Sechskant-schrauben Norm	Sechskant-schrauben Norm	Sechskant-schrauben Norm	Sechskant-schrauben Norm	Sechskant-schrauben Norm
Festigkeit	4.6/ 8.8	4.6/ 5.6/8.8/12.9	14H/22H	5.6/8.8	NA
WerkstoffNr	NA	NA	NA	NA	1.7709, 1.4301, 1.4986, 1.7258, 1.7711, 1.4571, 1.4541, 1.4725, 1.5406

Tabelle 4.2.: Evaluation: DBScan mit Merkmalsauswahl über Filter

Dieses beispielhaft dargestellte Muster konnte auch bei anderen Clustern erkannt werden. Die Entstehung der Cluster konnte also durchaus nachvollzogen werden. Jedoch trübte das Cluster 0, dass die als Rauschen gekennzeichneten Datensätze enthält, das Ergebnis. Die-

sem Cluster wurden fast ein Drittel der Daten zugeordnet, wodurch sich auch der schlechte Silhouetten Koeffizient erklären lässt. Aufgrund dem sehr hohen Anteil der Datensätze, die keinem Cluster zugeordnet werden konnten, ist dieses Resultat nicht akzeptabel und wurde vom Experten als nicht verwertbar bewertet.

4.1.2. Merkmalsauswahl über Wrapper

Clusterbildung mit k-Means

Bei der Clusterbildung mit der Merkmalsauswahl über den Wrapper und dem *k-Means*-Algorithmus wurde eine Clusteranzahl k von 28 ermittelt. Der Silhouetten Koeffizient dieser Clusterbildung beträgt 0.6500901 und besagt, dass eine brauchbare Struktur gefunden wurde. Das Resultat ähnelt dem aus Abschnitt 4.1.1. Auch hier wurden Cluster gebildet, die verständlich und sinnvoll erscheinen, wie das Cluster 5, das ausschließlich Gewindestifte enthält. Jedoch entstanden auch mit dieser Variante weitere Cluster, die Gewindestiften enthalten. Es ergab sich also ein zu dem in Abschnitt 4.1.1 vom Muster her gleiches Resultat. Die Datensätze des gleichen Clusters unterscheiden sich in den Merkmalen *NORM alpha*, *Festigkeit*, *WerkstoffNr* und *Werkstoff*. Dagegen wurden wiederum Datensätze, die sich nur in dem Merkmal *Festigkeit* unterscheiden, einem anderen Cluster zugeordnet. Da sich das Muster der Resultate gleicht, fällt auch die Bewertung ähnlich aus. Die entstandenen Cluster sind in ihrer Entstehung nicht nachvollziehbar und weisen keine verständlichen Strukturen auf, weshalb das Resultat nicht verwertbar ist.

Clusterbildung mit DBScan

Der *DBScan*-Algorithmus wurde mit den Parameter $\epsilon = 0.04$ und $minPoints = 30$ initial konfiguriert. Dabei entstanden 22 Cluster und ein Silhouetten Koeffizient von 0.5081363. Der Koeffizient beschreibt den Fund einer schwachen Struktur. Auch hier ist im Vergleich zur Merkmalsauswahl über die Filter ein sehr ähnliches Resultat entstanden, denn auch hier umfassen die als Rauschen gekennzeichneten Daten ungefähr ein Drittel der Datenmenge. Ebenfalls zeichnet sich dieses Resultat durch Cluster aus, die primär nach dem Merkmal *NORM alpha* gruppiert und sekundär nach den Merkmalen *ABVOST alpha*, *SICHNA alpha*, *OBFSTS alpha*, *Festigkeit*, *WerkstoffNr* und *Werkstoff* weiter untergliedert sind. Durch die gleichen Muster, die im Vergleich zu dem Resultat der Merkmalsauswahl über die Filter identifiziert wurden, fällt die Bewertung hier ähnlich aus. Die Cluster, ausgenommen dem Cluster 0, sind verständlich, jedoch ist dieses Resultat ebenfalls aufgrund des hohen Anteils der als Rauschen gekennzeichneten Datensätze unbrauchbar.

4.2. Durchlauf 2 bis N

Aus der Bewertung der vorherigen Resultate ging hervor, dass die Merkmale *SICHNA alpha* und *ABVOST alpha* noch synonyme Werte enthielten. Diese wurden vor Ausführung der folgenden Durchläufe bereinigt. Des Weiteren werden in diesem Abschnitt mehrere Durchläufe mit verschiedenen Konfigurationen zusammengefasst und dabei das aus Sicht des Experten beste Ergebnis dargestellt.

4.2.1. Merkmalsauswahl über Filter

Clusterbildung mit k-Means

Aufgrund der Unverständlichkeit des ersten Resultats, wurde nun versucht durch verschiedene Gewichtungen der Merkmale ein verbessertes Resultat zu erzielen. Für eine Vergleichbarkeit zum ersten Resultat bleibt die Clusteranzahl mit $k = 20$ zunächst unverändert. Nach Rücksprache mit dem Experten sollte dem Merkmal *NORM alpha* eine größere Bedeutung zukommen. Merkmale wie *OBFSTS alpha* und *WerkstoffNr* sollten hingegen weniger Einfluss auf das Resultat nehmen. Unter diesen Voraussetzungen wurde in mehreren Durchläufen mit verschieden gewichteten Merkmalen experimentiert. Das aus Expertensicht beste Ergebnis resultierte durch Gewichtung des Merkmals *NORM alpha* mit 50, wobei alle weiteren Merkmale das Standardgewicht von 1 behielten. Bei Betrachtung dieses Resultats fiel auf, dass zwei komplett identische Datensätze unterschiedlichen Clustern zugeordnet wurden. Der einzige Unterschied war im Merkmal *KOPFHH numer.* erkennbar, da ein Wert mit *12,5* und einer mit *12,50* angegeben war. Daraufhin stellte sich heraus, dass dieses Merkmal alphanumerische Werte enthielt, wodurch bei einer Kodierung dieser Werte dementsprechend die Unterschiede entstanden sind. Da die alphanumerischen Werte in diesem Merkmal nach Rücksprache mit dem Experten als fehlerhaft bewertet wurden, wurden diese zunächst bereinigt. Daraufhin wurde mit gleicher Gewichtung der Merkmale ein verschlechtertes Resultat erzielt. Durch Erhöhung des Gewichtes des Merkmals *NORM alpha* auf 100 wurde ein zur vorherigen Clusterbildung ähnliches Resultat erzielt. Die daraus resultierenden Cluster scheinen auf unterschiedliche Weisen entstanden zu sein. Durch das erhöhte Gewicht des Merkmals *NORM alpha* ist definitiv erkennbar, dass dieses Merkmal einen größeren Einfluss auf das Resultat hat, da viele dieser Cluster aus einem einzigen Typ Norm bestehen. Jedoch wurden auch mehrere Schraubentypen, wie etwa die Halbrundschrauben, T-Nutenschrauben und Linsen-Senkschrauben, in einem Cluster zusammengefasst. Zudem existiert beispielsweise noch der Schraubentyp der Sechskantschraube, der sich über 5 Cluster verteilt. Für die Entstehung dieser 5 Cluster waren neben dem Merkmal *NORM alpha* offensichtlich die Merkmale *ABVOST alpha* und *SICHNA alpha*, wie in Tabelle 4.3

an den Datensätzen 1 und 2 erkennbar, verantwortlich. Zusätzlich entstanden weitere Untergruppierungen nach den Merkmalen *OBFSTS alpha*, *Festigkeit* und *WerkstoffNr*, die in dieser Tabelle nicht dargestellt sind. Bei einzelner Betrachtung der Cluster waren diese verständlich, doch gesamtheitlich betrachtet in ihrer Entstehung sehr uneinheitlich. Die Datensätze 1 und 2 der Sechskantschrauben in Tabelle 4.3 unterscheiden sich, wie bereits erwähnt, in dem Merkmal *ABVOST alpha* und wurden daraufhin verschiedenen Clustern zugeordnet. Dieses Verhalten erwartete der Experte auch bei anderen Schraubentypen wie den Datensätzen 3 und 4 der Stiftschrauben, die jedoch dem gleichen Cluster zugeteilt wurden.

	Datensatz 1	Datensatz 2	Datensatz 3	Datensatz 4
SCHLUW numer.	24,0	24,0	-1,0	-1,0
KOPFHH numer.	10,0	10,0	-1,0	-1,0
ABVOST alpha	nein	ja	ja	nein
SICHNA alpha	nein	nein	nein	nein
KOPFBR numer.	-1,0	-1,0	-1,0	-1,0
GEWLG1 numer.	-1,0	-1,0	25,0	24,0
OBFSTS alpha	galv.verz.	galv.verz.	galv.verz.	galv.verz.
NORM alpha	Sechskant-schrauben Norm	Sechskant-schrauben Norm	Stiftschrauben Norm	Stiftschrauben Norm
Festigkeit	4.6	4.6	5.6	5.6
WerkstoffNr	NA	NA	NA	NA
Cluster	13	17	9	9

Tabelle 4.3.: Evaluation: k-Means mit Merkmalsauswahl über Filter mit veränderten Gewichten der Merkmale

Im Vergleich zum vorherigen Resultat aus Abschnitt 4.1.1 ist ein verbessertes Ergebnis entstanden, dass aber aufgrund der uneinheitlich entstanden Cluster für den Experten nicht sinnvoll ist. Der Grund für die uneinheitlich entstandenen Cluster könnte in der vorgegebenen Clusteranzahl liegen, die für eine Vergleichbarkeit zum ersten Resultat erhalten blieb. Jedoch werden dadurch unter Umständen Cluster erzwungen oder Cluster zusammengefasst. Daher wurde in einem weiteren Durchlauf eine neue Clusteranzahl k ermittelt. Mit gleichbleibender Gewichtung des Merkmals *NORM alpha = 100* wurde eine Clusteranzahl von 10 ermittelt.

Daraus ergab sich ein nahezu perfekter Silhouetten-Koeffizient von 0.9583829. Mit der Veränderung der Clusteranzahl wurden beispielsweise die Cluster der Sechskantschrauben, die sich über mehrere Cluster verteilten, zu einem Cluster zusammengefasst. Das bestätigt die Vermutung der erzwungenen Cluster. Die nun vorliegenden Cluster sind ebenfalls verständlich und nach dem Merkmal *NORM alpha* gruppiert. Jedoch werden in einigen Clustern nach wie vor mehrere Schraubentypen, wie die bereits erwähnten Schraubentypen der Halbrundschraube, T-Nutenschraube und Linsen-Senkschraube, zusammengefasst. Die zusammengefassten Schraubentypen sind nach Meinung des Experten inhaltlich nicht sinnvoll, zudem die Zusammenfassung willkürlich wirkt. Daher konnte auch durch die vorgenommenen Gewichtungen der Merkmale kein zufriedenstellendes Ergebnis erzielt werden.

Clusterbildung mit DBScan

Das erste Resultat des *DBScan*-Algorithmus war, vor allem aufgrund des hohen Anteils der als Rauschen gekennzeichneter Datensätze, unbrauchbar. Daher soll bei diesem Verfahren durch verschiedene Parameter Konfigurationen versucht werden, das Rauschen zu reduzieren, ohne dabei die Güte der Cluster zu verschlechtern. Mit den Einstellungen $\epsilon = 0,11$ und *minPoints = 2* wurden 100 Cluster erstellt. Dabei wurden nur 77 Datensätze als Rauschen gekennzeichnet und ein im Vergleich zum vorherigen Resultat aus Abschnitt 4.1.1 verbesserter Silhouetten-Koeffizient von 0.3880235 erreicht. Die entstandenen Cluster sind klar strukturiert und bestehen mit Ausnahme von zwei Clustern ausschließlich aus einem Typ Norm. Das Merkmal *NORM alpha* hat nach wie vor einen dominierenden Einfluss auf die Entstehung der Cluster. Jedoch haben auf unterschiedliche Cluster unterschiedliche Merkmale einen Einfluss auf die Clusterbildung, wie beispielsweise bei den Clustern der Gewindestifte und der Flügelschrauben. Dieses Verhalten ist durchaus nachvollziehbar, denn beim Typ der Gewindestifte sind die Unterschiede in den Merkmalen *OBFSTS alpha*, *Festigkeit* und *WerkstoffNr* auszumachen, wohingegen der Typ der Flügelschraube in diesen Merkmalen identische Werte aufweist und sich ausschließlich im Merkmal *KOPFHH numer.* unterscheidet. Die Entstehung der Cluster ist nachvollziehbar und die Cluster sind in sich verständlich. Dieses Ergebnis liefert jedoch nach Meinung des Experten eine zu hohe Anzahl von Clustern. Ebenso wäre es wünschenswert, dass eine Gruppierung nach einheitlichen Kriterien entsteht. Mit dem hier erzielten Resultat konnte zumindest dem Problem des hohen Anteils, der als Rauschen gekennzeichneter Datensätze, entgegengewirkt werden. Die vielen entstandenen Cluster konnten durch angepasste Parameter-Einstellungen zwar verringert werden, doch waren die daraus resultierenden Cluster nicht verständlich.

4.2.2. Merkmalsauswahl über Wrapper

Clusterbildung mit k-Means

Auf Basis des ersten Resultats, welches als unbrauchbar bewertet wurde, soll nun mit Hilfe von Gewichtungen der Merkmale ein verbessertes Resultat erzielt werden. Um eine Vergleichbarkeit der Resultate zu schaffen, bleibt auch hier, wie bereits bei der Merkmalsauswahl über Filter, die Clusteranzahl k mit dem Wert 28 aus dem ersten Durchlauf bestehen. Nachdem mit einigen Gewichtungen der Merkmale experimentiert wurde, wurde das Resultat mit den Gewichtungen für das Merkmal *GEWDUR alpha = 10* und *NORM alpha = 200* als bestes bewertet. Der aus dem Resultat ermittelte Silhouetten Koeffizient verbesserte sich auf 0.7246698. In den Clustern befindet sich, bis auf wenige Ausnahmen, ausschließlich ein Typ von Schrauben. Einige Schraubentypen erstrecken sich über mehrere Cluster, wie beispielsweise der Typ der Zylinderschraube. Die Zylinderschraube ist in 3 Clustern vertreten, die nach den Merkmalen *ABVOST alpha* und *SICHNA alpha* gruppiert sind. Die Sechskantschrauben sind hingegen in 9 Clustern vertreten und in einer etwas anderen Art gruppiert, die nachfolgend erläutert wird. Zunächst besteht eines dieser Cluster nur aus Sechskantschrauben mit *GEWDUR alpha = Metrisch mit Steigung*. Die restlichen 8 Cluster der Sechskantschrauben besitzen den Wert *Metrisch*. Diese sind weiter nach den Merkmalen *ABVOST alpha* und *SICHNA alpha* unterteilt. Eines der Cluster enthält die Datensätze, deren Merkmalswerte jeweils *ja* besitzen. Einem weiteren Cluster sind die Datensätze, deren Merkmalswerte jeweils *nein* aufweisen, zugeordnet. Die 6 weiteren Cluster besitzen den Wert *ja* für das Merkmal *ABVOST alpha* und den Wert *nein* für das Merkmal *SICHNA alpha* und untergliedern sich weiter nach den Merkmalen *OBFSTS alpha*, *Festigkeit* und *Werkstoff*. Die Cluster sind einzeln betrachtet verständlich, jedoch gesamtheitlich gesehen uneinheitlich entstanden. Bei Betrachtung der Datensätze 1 und 3 der Sechskantschrauben in Tabelle 4.4, ist der Unterschied dieser in den Merkmalen *WerkstoffNr* und *Werkstoff* erkennbar, wodurch diese verschiedenen Clustern zugeordnet wurden. Die Datensätze 2 und 4 bilden ein Abbild der Datensätze 1 und 3 mit dem Unterschied, dass diese Datensätze vom Typ der Zylinderschraube sind. Die Erwartung des Experten war, dass diese Datensätze, gemäß der Clusterbildung der Sechskantschrauben, ebenfalls verschiedenen Clustern zugeordnet werden. Entgegen der Erwartungen wurden diese Datensätze aber dem gleichen Cluster zugewiesen.

	Datensatz 1	Datensatz 2	Datensatz 3	Datensatz 4
GEWDUR alpha	Metrisch	Metrisch	Metrisch	Metrisch
ABVOST alpha	ja	ja	ja	ja
SICHNA alpha	nein	nein	nein	nein
GEWLG1 numer.	-1,0	-1,0	-1,0	-1,0
FORM alpha	NA	NA	NA	NA
OBFSTS alpha	blank	blank	galv.verz.	galv.verz.
NORM alpha	Sechskant-schrauben Norm	Zylinder-schrauben Norm	Sechskant-schrauben Norm	Zylinder-schrauben Norm
Festigkeit	NA	NA	5.6	5.6
WerkstoffNr	1.4541	1.4541	NA	NA
Werkstoff	X6CrNiTi18-10	X6CrNiTi18-10	NA	NA
GEWENDEN	NA/NA	NA/NA	NA/NA	NA/NA
Cluster	25	8	26	8

Tabelle 4.4.: Evaluation: k-Means mit Merkmalsauswahl über Wrapper und veränderten Gewichten der Merkmale

Die Antwort in diesem Verhalten könnte in der vorgegeben Clusteranzahl von k mit 28 Gruppen liegen, wodurch Cluster erzwungen oder zusammengefasst werden könnten. Auf Basis dessen wurde eine neue Clusteranzahl von 8 Clustern ermittelt, mit denen ein Silhouetten-Koeffizient von 0.9170567 erreicht wurde. Dieses Resultat fasst hingegen viele Schraubentypen zusammen. Einzig die Schraubentypen der Sechskantschrauben, Zylinderschrauben und Gewindestifte bildeten jeweils ein eigenes Cluster. Im Gegensatz zum vorherigen Resultat haben die Merkmale *ABVOST alpha* und *SICHNA alpha*, sowie *OBFSTS alpha*, *Festigkeit* und *Werkstoff* keinen Einfluss auf die Gruppierungen, sodass sich die Vermutung der erzwungenen Cluster bestätigt hat. Die Zusammenstellung der Schraubentypen innerhalb der Cluster erfolgte nach Meinung des Experten willkürlich und ohne ersichtliche Systematik. Ein weiterer Grund dafür, dass der Experte das Resultat als nicht nützlich bewertete, war die Gruppierung von inhaltlich nicht zueinander passenden Schraubentypen.

Clusterbildung mit DBScan

Ebenfalls, wie in der Merkmalsauswahl über Filter, gilt es mit der Merkmalsauswahl über den Wrapper in den Folgedurchläufen das Rauschen durch verschiedene Parameter Einstellungen

zu verringern, ohne dabei die Güte der Cluster zu verschlechtern. Mit den Einstellungen $\epsilon = 0,05$ und *minPoints = 2* wurden 160 Cluster erstellt, wobei 104 Datensätze als Rauschen gekennzeichnet wurden. Mit diesem Resultat wurde ein Silhouetten-Koeffizient von 0.8869057 erzielt. Das Ziel der Verringerung des Rauschens ohne Verschlechterung der Güte der Cluster wurde erreicht. Die Cluster sind verständlich und primär nach dem Merkmal *NORM alpha* entstanden. Die *Zylinderschrauben Norm* erstreckt sich beispielsweise über 23 Cluster, in denen die Zylinderschrauben nach den Merkmalen *GEWDUR alpha*, *ABVOST alpha*, *SICHNA alpha*, *OBFSTS alpha*, *Festigkeit* und *Werkstoff* weiter unterteilt sind. In der Tabelle 4.5 sind einige Cluster dargestellt, die der Veranschaulichung des Resultats dienen sollen.

Cluster	31	41	32	33
GEWDUR alpha	Metrisch	Metrisch	Metrisch	Metrisch
ABVOST alpha	ja	ja	ja	ja
SICHNA alpha	nein	nein	ja	nein
GEWLG1 numer.	-1	-1	-1	-1
FORM alpha	NA	NA	NA	NA
OBFSTS alpha	galv.verz.	galv.verz.	galv.verz.	blank
NORM alpha	Zylinder-schrauben Norm	Zylinder-schrauben Norm	Zylinder-schrauben Norm	Zylinder-schrauben Norm
Festigkeit	8.8	5.6	5.6	NA
WerkstoffNr	NA	NA	NA	1.4301, 1.4541, 1.4571
Werkstoff	NA	NA	NA	X5CrNi18-10, X5CrNi18-10, X6CrNiMoTi17-12-2
GEWENDEN	NA/NA	NA/NA	NA/NA	NA/NA

Tabelle 4.5.: Evaluation: DBScan mit Merkmalsauswahl über Wrapper und veränderten Parametern

Die Cluster 31 und 41 unterscheiden sich beispielsweise in dem Merkmal *Festigkeit*, wohingegen sich das Cluster 32 im Merkmal *SICHNA alpha* zum Cluster 41 abgrenzt. Diese Muster konnten auch bei den anderen Schraubentypen entdeckt werden, wodurch verständliche Gruppierungen nach einem einheitlichen Schema entstanden sind. Einzig mit dem Umstand der

sehr vielen Cluster war der Experte nicht zufrieden, welches nicht durch andere Parameter Einstellungen zufriedenstellend gelöst werden konnte.

4.3. Finaler Durchlauf

4.3.1. Merkmalsauswahl über Filter

Clusterbildung mit k-Means

Da allein durch eine Veränderung der Gewichte kein zufriedenstellendes Resultat erzielt werden konnte, sollen nun Merkmale aus der Merkmalsmenge entfernt werden. Dafür werden die zur Elimination auserwählten Merkmale mit einer Gewichtung von 0 versehen. Zudem wird für jede verminderte Merkmalsmenge eine neue optimale Clusteranzahl k ermittelt, um der Problematik der erzwungenen bzw. zusammengefassten Cluster entgegenzuwirken. Die Elimination eines geeigneten Merkmals wurde mit Hilfe des Experten durchgeführt. Dazu wurde in jedem Durchgang das Merkmal identifiziert und eliminiert, dem nach Meinung des Experten die geringste Bedeutung zukommt. Das wurde wiederholt, bis ein zufriedenstellendes Ergebnis vorlag oder alle Merkmale entfernt wurden. Unberücksichtigt blieb dabei, dass beispielsweise Merkmale korrelieren können und durch Elimination eines der korrelierenden Merkmale ein verschlechtertes Resultat erzielt werden könnte. Ebenso ist es möglich, dass der Experte bei der Identifizierung des Merkmals mit der geringsten Bedeutung daneben liegt. Bei Berücksichtigung aller möglichen Problematiken müssten im schlechtesten Fall alle möglichen Unterräume bewertet werden. Bei 10 Merkmalen würden also im schlechtesten Fall $2^{10} - 1 = 1023$ Cluster-Analysen durchgeführt und bewertet werden müssen. Daher wurde einzig auf die Meinung des Experten vertraut und schrittweise ein Merkmal aus der Menge der Merkmale entfernt. Im Folgenden ist die Reihenfolge der Merkmals-Elimination dargestellt:

1. SCHLUW numer.
2. KOPFHH numer.
3. KOPFBR numer.
4. GEWLG1 numer.
5. Werkstoff
6. Festigkeit
7. OBFSTS alpha
8. SICHNA alpha
9. ABVOST alpha

Anhand des Silhouetten Koeffizienten wurde bei Elimination der Merkmale *KOPFHH numer.* und *KOPFBR numer.* eine Verschlechterung des Resultats identifiziert, die sich jedoch nicht in der Bewertung des Experten widerspiegelte. In allen Ergebnissen wurden verschiedene Schraubentypen in Clustern zusammengefasst, wobei die Auswahl der Schraubentypen innerhalb der verschiedenen Resultate zum Teil variierten. Eine Verschlechterung des Resultats trat nach Meinung des Experten nur nach der Entfernung des Merkmals *Werkstoff* auf. Begründet

wurde diese Verschlechterung in einem Anstieg der Clusteranzahl auf 56 Cluster. Jedoch verbesserte sich das Resultat mit Elimination der weiteren Merkmale stetig. Letztendlich wurden alle Merkmale bis auf *NORM alpha* aus der Menge der Merkmale entfernt, bis das Ergebnis zufriedenstellend war. Daraus resultierten 20 Cluster mit einem Silhouetten Koeffizienten von 0.999684. Jedes dieser Cluster besteht aus einem Typ von Schrauben und gruppiert diese nach dem Merkmal *NORM alpha* wie folgt:

1. Flachkopfschrauben Norm
2. Halbrundschrauben Norm
3. Verschlussschrauben Norm
4. T-Nutenschrauben Norm
5. Linsen-Senkschrauben Norm
6. Gewindestift Norm
7. Spannschrauben Norm
8. Holzschrauben Norm
9. Stiftschrauben Norm
10. Gewinde-Schneidschrauben Norm
11. Steinschrauben Norm
12. Flachrundschrauben Norm
13. Flügelschrauben Norm
14. Hammerschrauben Norm
15. Tellerschrauben Norm
16. Senkschrauben Norm
17. Kreuzgriffschrauben Norm
18. Vierkantschrauben Norm
19. Zylinderschrauben Norm
20. Sechskantschrauben Norm

Im Gegensatz zu den vorherigen Ergebnissen existieren keine Cluster, die einen Mix von Schraubentypen, wie etwa Holzschrauben und Senkschrauben, enthalten, weswegen die Resultate oftmals als unbrauchbar bewertet wurden. Das nun erzielte Resultat liefert verständliche Gruppen, die mehrere Normen eines Schraubentypen zusammenfassen, die bei bestehender Klassifizierung jeweils eine eigene Klasse bildeten. Dieses Ergebnis hat den Experten im Bezug auf den Gegenstandsbereich der Schrauben sensibilisiert und eine andere Möglichkeit der Gruppierung aufgezeigt. Dem Experten genügt dieses Resultat hinsichtlich der Kriterien der Verständlichkeit, Neuartigkeit, sowie Nutzbarkeit, weswegen dieses Resultat eine zufriedenstellende Lösung bietet.

Clusterbildung mit DBScan

Im vorherigen Durchlauf konnte durch eine angepasste Parameter Konfiguration bereits eine Reduzierung des Rauschens erzielt werden. Das entstandene Resultat wurde allerdings aufgrund der sehr hohen Anzahl von Clustern als nicht nutzbar bewertet. Durch Gewichtungen von Merkmalen soll nun versucht werden viele dieser Cluster zusammenzufassen. Dazu ist es neben der Gewichtung von Merkmalen ebenfalls nötig eine Anpassung an den Parametern vorzunehmen. Durch eine höhere Gewichtung eines Merkmals sollen größere Distanzen zwischen Objekten, die sich in diesem Merkmal ungleich sind, erreicht werden. Ebenso können sich

Objekte, die sich in den gewichteten Merkmalen gleichen, in weniger bedeutsamen Merkmalen unterscheiden können, ohne gleich aus der ϵ-Umgebung zu fallen. In den folgenden Durchläufen wurde also sowohl mit verschiedenen Gewichtungen von Merkmalen, als auch der damit einhergehenden Anpassungen der Parameter experimentiert. Mit den Einstellungen $\epsilon = 1.2$, $minPoints = 2$ und einer Gewichtung des Merkmals *NORM alpha = 50* wurde ein zum in Abschnitt 4.3.1 identisches Resultat erzielt. Einzig ein Datensatz der Kreuzgriffschraube wurde als Rauschen gekennzeichnet. Da die Kreuzgriffschraube der einzige Datensatz dieses Schraubentyps ist und der Parameter $minPoints$ mit 2 gewählt wurde, kann dieser Datensatz kein eigenes Cluster bilden. Wird der Parameter $minPoints$ auf 1 herab gesetzt, bilden womöglich alle als Rauschen gekennzeichneten Datensätze ein eigenes Cluster, was im vorliegenden Fall mit einem Datensatz kein Problem darstellt. Handelt es sich dagegen um mehrere als Rauschen gekennzeichneter Datensätze, könnte dies zu einem ungewollten Resultat führen. Der Versuch mittels anderer Gewichtungen ein alternatives Resultat zu erzielten, das ebenfalls den Kriterien der Bewertung genügt, war nicht erfolgreich.

4.3.2. Merkmalsauswahl über Wrapper

Clusterbildung mit k-Means

Im Abschnitt 4.3.1 wurde bereits durch Elimination von Merkmalen versucht ein gutes Resultat zu erzielen. Das gleiche Vorgehen wird nun hier mit der Merkmalsauswahl über den Wrapper angewandt. Zunächst soll auch hier die Reihenfolge der Merkmals-Elimination auf Basis des Expertenwissens veranschaulicht werden.

1. FORM alpha
2. GEWLG1 numer.
3. Werkstoff
4. WerkstoffNr
5. Festigkeit
6. OBFSTS alpha
7. SICHNA alpha
8. ABVOST alpha
9. GEWDUR alpha
10. GEWENDEN

Identisch zur Elimination, der über die Filter ermittelten Merkmale, wurden alle Merkmale bis auf das Merkmal *NORM alpha* aus der Merkmalsmenge entfernt. Eine vom Experten bewertete Verschlechterung des Resultats trat nach Entfernung des Merkmals *Festigkeit* auf, was durch eine hohe Anzahl von Clustern begründet wurde. Mit Elimination der Folgemerkmale wurde das Resultat allerdings wieder stetig besser. Da die Merkmalsmenge auf dieselbe wie im Abschnitt 4.3.1 reduziert wurde, ist es wenig verwunderlich, dass hier das gleiche Resultat entstand.

Clusterbildung mit DBScan

Ebenfalls, wie bereits bei der Merkmalsauswahl über Filter, wird durch eine Veränderung der Gewichtungen der Merkmale und damit einhergehender Anpassung der Parameter des *DBScan*-Algorithmus versucht die entstandenen Cluster aus Abschnitt 4.2.2 zusammenzufassen. Mit der Gewichtung des Merkmals *NORM alpha = 50* und den Parameter-Einstellungen $\epsilon = 1.5$ und $minPoints = 1$ konnte das bereits als sinnvoll evaluierte Resultat aus den vorherigen Abschnitten erreicht werden. Dieses Resultat erzielte einen Silhouetten Koeffizienten von 0.8034135. Des Weiteren wurde auch hier versucht, durch verschiedene Gewichtungen der Merkmale und Parameter-Einstellungen, ein alternatives Ergebnis zu erzielen. Durch die Gewichtung des Merkmals *GEWDUR alpha = 10* und gleichzeitiger Herabsetzung des Gewichtes von *NORM alpha* auf 1, sowie unveränderten Parametern, konnte ein weiteres erwähnenswertes Resultat erzielt werden. In diesem Resultat entstanden 8 Cluster, die jeweils einen Gewindetypen repräsentierten und wie folgt aussehen:

1. Metrisches Gewinde
2. Metrisches Gewinde mit Steigung
3. Zwei Gewindeenden
4. Zylindrisches Gewinde
5. Kegelige Gewinde
6. Holzschraubengewinde
7. Panzergewinde
8. Zölliges Gewinde

Durch dieses Resultat wurde laut dem Experten eine sehr interessante Erkenntnis errungen. Diese Art von Gruppierung bietet eine völlig neue Sichtweise, die durchaus den Kriterien der Bewertung genügt. Jedoch wurde das Resultat aus Abschnitt 4.3.1, dem hier erzielten, vorgezogen.

4.4. Gegenüberstellung

Aus den vorherigen Abschnitten ist zu entnehmen, dass sowohl durch beide Clustering-Verfahren als auch mit beiden Merkmalsmengen dasselbe Resultat erzielt werden konnte. Zunächst sollen die unterschiedlichen Merkmalsmengen mit der Verwendung des *DBScan*-Algorithmus verglichen werden, da dort keine Elimination von Merkmalen zur Erzielung eines verwertbaren Resultats erforderlich war. Die Möglichkeit zur Erreichung desselben Resultats ist nur durch die Existenz der für das Resultat bedeutsamen Merkmale in beiden Merkmalsmengen gegeben. Wären die entscheidenden oder das entscheidende Merkmal in der Merkmalsmenge nicht enthalten, ist es durchaus unwahrscheinlich, dass ein identisches Resultat erzielt werden kann. Denkbar wäre hingegen, dass auf Basis anderer Merkmale zufällig das gleiche Resultat entstehen könnte. Da mit beiden Merkmalsmengen identische Cluster entstanden sind, kann allein durch die Betrachtung der Resultate keine Bewertung bezüglich des besseren Verfahrens

bzw. der besseren Merkmalsmenge vorgenommen werden. Jedoch kann festgehalten werden, dass beide Verfahren der Attribut-Selektion hinsichtlich einer erfolgreichen Cluster-Analyse geeignete Merkmalsmengen ermittelten. Für eine Bewertung hinsichtlich der qualitativ besseren Merkmalsmenge wird der Silhouetten Koeffizient herangezogen. Mit der Merkmalsmenge des Wrappers wurde ein Resultat mit einem Koeffizienten von 0.8034135 erzielt. Dem gegenüber steht die Merkmalsmenge der Filter dessen Resultat mit einem Koeffizienten von 0.7921964 das qualitativ schlechtere darstellt. Das bedeutet, dass durch den Wrapper eine Merkmalsmenge selektiert wurde, die besser gruppierbar als die der Filter ist. Das sagt jedoch nicht aus, dass über den Wrapper grundsätzlich die bessere Merkmalsmenge ermittelt wird. Denn wie eingangs erwähnt ist es durchaus möglich, dass eines der Verfahren Merkmale ermittelt durch die ein Resultat erzielt wird, welches durch die ermittelte Merkmalsmenge des jeweils anderen Verfahrens nicht erzielbar ist.
Neben den unterschiedlichen Verfahren zur Merkmalsauswahl wurden auch unterschiedliche Clustering-Verfahren eingesetzt. Hier beschränkt sich der Vergleich auf eine gemeinsame Merkmalsmenge, wie die über den Wrapper selektierte, um eine Vergleichbarkeit der Verfahren zu schaffen. Auch bei den Clustering-Verfahren war es möglich dasselbe Resultat zu erzielen, wofür verschiedene Maßnahmen für die verschiedenen Verfahren nötig waren. Doch auf Basis der Erzielung desselben Resultats kann bereits festgehalten werden, dass beide Verfahren für ein Clustering im Kontext technischer Artefakte geeignet waren. Während bei der Verwendung des *DBScan*-Algorithmus lediglich eine Gewichtung der Merkmale vorgenommen und mit unterschiedlichen Parameter Konfigurationen experimentiert wurde, war es bei der Verwendung des *k-Means*-Algorithmus zusätzlich nötig die Merkmalsmenge zu reduzieren. Da bei der Elimination von Merkmalen, wie in Abschnitt 4.3.1 beschrieben, durchaus Probleme auftreten können und unter Umständen nicht die richtigen Merkmale aus der Merkmalsmenge entfernt werden, ist die Verwendung des *DBScan*-Algorithmus für diesen Gegenstandsbereich weniger aufwendig und fehlerträchtig. Ebenso wurden durch den *DBScan*-Algorithmus im Gegensatz zu dem *k-Means*-Algorithmus mit jedem Durchlauf Ergebnisse erzielt, die verständlich waren. Eine Bewertung des besseren Verfahrens auf Basis des Silhouetten Koeffizienten ist wegen der unterschiedlichen Merkmalsmengen, die in die Analyse einflossen, wenig aussagekräftig. Da bei der Cluster-Analyse mit *k-Means* nur ein Merkmal bei der Analyse berücksichtigt wurde, konnte bei diesem Resultat ein nahezu perfekter Silhouetten Koeffizient von 0.999684 erzielt werden. Im Gegensatz dazu flossen bei der Cluster-Analyse mit *DBScan* alle selektierten Merkmale in die Analyse ein und es wurde ein Koeffizient von 0.8034135 erreicht. Der schlechtere Koeffizient ist eben auf Basis der unterschiedlichen Merkmalsmengen zu begründen. Denn bei einer Cluster-Analyse unter Berücksichtigung der gesamten Merkmalsmenge

differenzieren sich Objekte innerhalb der Cluster in einigen Merkmalen, wobei dies unter Berücksichtigung eines einzigen Merkmals nicht der Fall ist. Da der *DBScan*-Algorithmus gegenüber dem *k-Means*-Algorithmus in den notwendigen Maßnahmen zur Erzielung eines verwertbaren Resultats weniger aufwendig war, sowie ein geringeres Fehler Potenzial beherbergte, ist dieser bei der Cluster-Analyse technischer Artefakte vorzuziehen. Zudem wurden durch den *DBScan*-Algorithmus stets verständliche Resultate erzeugt.
Zusammenfassend kann festhalten werden, dass die im Rahmen dieser Arbeit verwendeten Verfahren für die Attribut-Selektion und Cluster-Analyse in Bezug auf die Gruppierung technischer Artefakte geeignet waren. Trotzdem sei an dieser Stelle darauf hingewiesen, dass andere Verfahren durchaus gleich oder sogar besser geeignet sein könnten.

5. Schluss

5.1. Fazit

Das Ziel dieser Arbeit lag in der Überprüfung des KDD-Prozesses in Bezug auf dessen Eignung im Kontext einer Cluster-Analyse technischer Artefakte. Dafür wurde der Prozess praktisch an dem Gegenstandsbereich der Schrauben durchgeführt. Zunächst bestand die Aufgabe in der Überprüfung der Strukturierung der durch das Unternehmen selektierten Daten hinsichtlich deren Eignung für die Cluster-Analyse. Bereits bei der Datenintegration wurden Schwächen in den Strukturen der Daten erkannt. Bevor die Datentabellen in einen zentralen Datenbestand überführt wurden, konnten durch die Analyse der Datentabellen synonyme Attribute identifiziert werden. Doch auch in den weiteren Phasen konnten Schwächen in den Strukturen und der Qualität der Daten festgestellt werden. Da wären synonyme Datenwerte für die Oberfläche einer Schraube, wie *feu.verz.*, *feuer_verz.*, *feuer.verz.* und *feuerverzinkt*. Bei einer Daten-Abfrage von Schrauben mit der Oberfläche *feuerverzinkt* würde das Resultat aufgrund dieser verschiedenen Werte nicht alle Schrauben enthalten, die diesem Wert semantisch entsprechen. Ebenso ist durch die Freitext-Eingabe die Verständlichkeit der Daten nicht gegeben, da unter anderem Kürzel Verwendung finden, die nicht für jeden verständlich sind. Auch die Korrektheit der Daten leidet unter der Freitext-Eingabe durch Informationen, die falschen Attributen zugeordnet werden. Nach der Identifizierung der Probleme in den Daten fand die Bereinigung statt. Des Weiteren wurde eine Daten- und Dimensionsreduktion durchgeführt. Ebenfalls war eine Kodierung und Normalisierung der Daten, um diese auf das Cluster-Analyse-Verfahren vorzubereiten, notwendig. Die durch die Cluster-Analyse erzielten Resultate wurden dem Fachgebietsexperten präsentiert, wodurch diesem verschiedene Betrachtungsweisen einer möglichen Gruppierung aufgezeigt wurden. Nach einigen Anpassungen und Iterationen des KDD-Prozesses entstand ein Resultat, dass die ungeordnete Datenmenge in homogene Teilmengen gruppierte, die als verständlich, neuartig und nützlich evaluiert wurden. Durch dieses Resultat fand eine erfolgreiche Sensibilisierung des Experten bezüglich der Gruppierung des Gegenstandsbereiches der Schrauben statt. Der KDD-Prozess hat sich also in der Eignung in Bezug auf das Clustering technischer Artefakte absolut bewährt. Besonders wird während

dieses Prozesses der Handlungsbedarf in den Strukturen der Daten durch die Aufdeckung der Schwachstellen sichtbar.

5.2. Ausblick

Der KDD-Prozess wurde in dieser Arbeit exemplarisch am Gegenstandsbereich der Schrauben durchgeführt und hat sich in der Eignung bewährt. Eine nächste Herausforderung stellt die Übertragbarkeit dieses Verfahrens auf ähnliche Bereiche, wie etwa Motoren, oder ein anderes Unternehmen dar.
Die Daten anderer Unternehmen, im Speziellen auch die der Schrauben, weisen sehr wahrscheinlich abweichende Strukturen zu denen, die dieser Arbeit zugrunde liegen, auf. Ebenso verhält es sich mit den Daten ähnlicher Bereiche. Seien es die Attribute, die für die Beschreibung eines Teiles genutzt werden, die verwendeten Datentypen oder die Datenwerte, es sind Anpassungen an die zugrundeliegenden Daten nötig. Dabei handelt es sich nicht nur um Anpassungen der Datenwerte, wie beispielsweise bei der Übersetzung von synonymen Datenwerten. Der Gegenstandsbereich der Motoren enthält beispielsweise Attribute für die Motorleistung in PS und kW, die semantisch identisch sind. Bei der Zusammenfassung dieser Attribute entstehen Datenwertkonflikte in Form unterschiedlicher Maßeinheiten, die auf eine gemeinsame Einheit, wie etwa kW, zu überführen sind. Da in den Daten der Schrauben keine Datenwertkonflikte auftraten, wurde die Problemlösung in dem Verfahren dieser Arbeit nicht berücksichtigt. Ebenso wurde im Zuge der Bereinigung falscher Daten ein speziell auf das vorliegende Problem entwickeltes Programm genutzt, welches für die Daten ähnlicher Bereiche oder anderer Unternehmen sehr wahrscheinlich unbrauchbar ist. An diesen Beispielen soll verdeutlicht werden, dass andere Daten andere Probleme mit sich bringen können. Ebenso können gleiche Probleme unterschiedliche Reaktionen erfordern. In dem Verfahren der Schrauben wurden beispielsweise die fehlenden Werte einiger Attribute durch eine globale Konstante ersetzt, während bei Daten anderer Gegenstandsbereiche oder Unternehmen die Bereinigung der fehlenden Daten durch den wahrscheinlichsten Wert oder einem Durchschnittswert als sinnvoll erscheinen könnte. Dadurch soll jedoch nicht zum Ausdruck kommen, dass das Verfahren in Bezug auf die Übertragbarkeit vollkommen unbrauchbar ist. Es wurden sehr wohl Lösungen erstellt, die Wiederverwendung finden können. Bei der Zusammenfassung von semantisch äquivalenten Attributen, der Übersetzung von Merkmalswerten gleicher Bedeutung oder der Bereinigung fehlender Werte durch eine globale Konstante oder über Relationen handelt es sich um Lösungen von Problemen, die durchaus in anderen Daten auftreten und dementsprechend wiederverwendet werden können. Gerade auch die Verfahren zur Kodierung, Normalisierung

und Attribut-Selektion werden immer wieder benötigt und können in ähnlichen Bereichen, sowie anderen Unternehmen wiederverwendet werden. Es handelt sich also um ein Verfahren, das Lösungen für immer wiederkehrende Probleme bereitstellt, jedoch nicht gänzlich übertragbar ist. Dennoch bietet das Verfahren Teillösungen, die auch für ähnliche Bereiche oder andere Unternehmen brauchbar sind.
Weiterführend besteht noch die Herausforderung der Verwendbarkeit des Verfahrens über den Bereich technischer Artefakte hinaus bzw. der Generalisierung. Beispielsweise könnte in einem Mode-Vertrieb die Aufgabe der Gruppierung von Kleidungsstücken vorliegen. Doch auch in den Daten von Kleidungsstücken können etwa für die Art des Kleidungsstückes synonyme Datenwerte wie *Hoodie*, *Hoody* und *Kapuzenpullover* auftreten. Ebenso könnte eine Komprimierung der Kleidungsstückarten, wie beispielsweise *Kapuzenpullover*, *Sweatshirt*, *Hemd* und *T-Shirt* zu *Oberteil*, nötig sein. Eine Kodierung, Normierung und Attribut-Selektion könnten ebenfalls erforderlich sein. Für all diese Probleme bietet das Verfahren dieser Arbeit Lösungen an, die auch in diesem Kontext genutzt werden können. Jedoch könnten unterschiedliche Attribute für die Größe der Kleidungsstücke bestehen. In einem Attribut könnte die Angabe der Konfektionsgröße in numerischer Form, wie *44*, *46*, *48*, *usw.*, erfolgen, in einem weiterem wiederum in textueller Form, wie *S*, *M*, *L*, *usw.* Bei der Zusammenführung der semantisch äquivalenten Attribute entstehen Datenwertkonflikte, die gelöst werden müssen. Es wurde bereits das gleiche Problem bei dem Gegenstandsbereich der Motoren identifiziert und festgestellt, dass das Verfahren dieser Arbeit keine Lösung dafür bereitstellt. Das am Gegenstandsbereich der Schrauben beschriebene Vorgehen bietet demnach Lösungen, die über den Bereich technischer Artefakte hinaus nutzbar sind. Dabei handelt es sich, wie bei den Überlegungen zur Übertragbarkeit, um Teillösungen des Verfahrens und nicht um das Verfahren an sich.
Zusammenfassend ist festzuhalten, dass es sich bei dem KDD um einen interaktiven und iterativen, aber keinen vollautomatischen Prozess handelt. Es gibt kein Patentrezept zur Durchführung des KDD-Prozesses, da dieser das individuelle Handeln des Anwenders hinsichtlich der zugrundeliegenden Daten erfordert.
Im Weiteren soll noch ein kurzer Ausblick hinsichtlich der Verwendung der gewonnenen Erkenntnisse resultierend aus der Durchführung einer Cluster-Analyse mittels dem KDD-Prozess folgen. Eine der Erkenntnisse könnte die Aufdeckung struktureller Schwachstellen in den Daten sein, wodurch eine Restrukturierung denkbar wäre. Weiter ist bei der Bildung von sinnvollen, verständlichen und nützlichen Gruppierungen eine weiterführende Aufgabe in Form eines Klassifikators denkbar, da die manuelle Zuordnung eines neuen Objekts zu einer Gruppe fehlerträchtig ist. Der Klassifikator hingegen könnte mit den aus der Cluster-Analyse entstandenen Gruppierungen trainiert werden, um zukünftig die Klassifizierung neuer

Objekte durchzuführen. Es kann also durchaus etwas mit den Erkenntnissen aus der Durchführung des KDD-Prozesses angefangen werden, unabhängig von der Bildung von sinnvollen, verständlichen und nützlichen Gruppierungen.

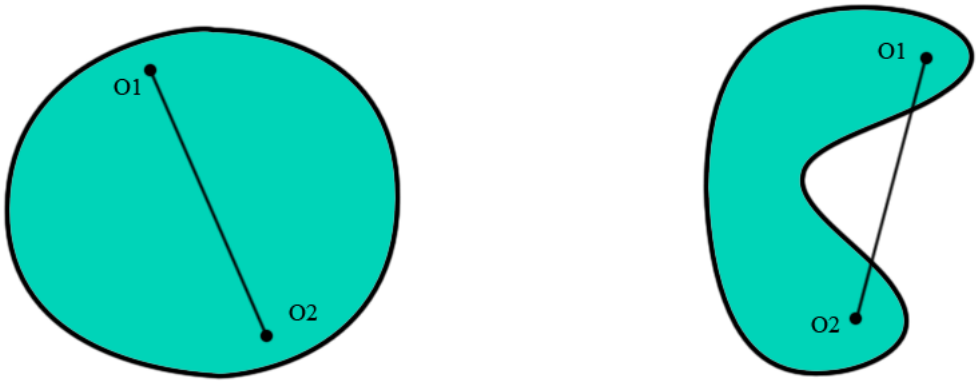
A. Anhang

A.1. Metadaten Schrauben

Merkmal	Beschreibung
Teile-Nr	Teile-Nr
Teilebezeichnung	Teilebezeichnung
GEWDUR alpha	Gewinde
LANGE numer.	Länge (mm)
FESTWE alpha	Festigkeit/Werkstoff
OBFBEH alpha	Art Oberfläche
STOFNR alpha	Werkstoff-Nummer
GEWLGE numer.	Länge Gewinde
SCHLUW numer.	Schlüsselweite SW
KOPFHH numer.	Höhe Kopf
ABVOST alpha	Art Abnahmevorschr.
SICHNA alpha	Sicherheitsnachweis
TEAB01 alpha	TempWarmstr.grenze 1
TEAB02 alpha	TempWarmstr.grenze 2
NENNLA numer.	Länge Nenn-
KOPFBR numer.	Durchmesser Kopf
ANSKUP alpha	Ausf.Ansatzkuppe
LIEUMF alpha	Lieferumfang
GEWEN1 alpha	Gewindeende 1
GEWLG1 numer.	Länge Gewinde 1
GEWEN2 alpha	Gewindeende 2
GEWLG2 numer.	Länge Gewinde 2
FORM alpha	Form
WERKST alpha	Werkstoff
KOPFBM numer.	Breite Kopfbreite M

OBFSTS alpha	Oberfläche Steinsch.
PRODKL alpha	Produktklasse
DURCHM numer.	Durchmesser
DUSCHF numer.	Durchmesser Schaft
NORM alpha	Norm
WERKSH alpha	Werkstoff Schraube
HERSTE alpha	Hersteller
SCHABR numer.	Breite Schaft
VERPEH alpha	Verpackungseinheit
SCHLFO alpha	Form Kreuzschlitz
FRM962 alpha	Form Schraube DIN962
AEENDE alpha	Ausf.Einschraubenden
FESTKL alpha	Festigkeit Schrauben
ANMUTE numer.	Anzahl Muttern/Teil
TELSCH alpha	Tellerscheibe
TECLIF alpha	Techn.Lieferbeding.
DUSCHB alpha	Durchmesser Scheibe
KKLASS alpha	K-Klasse
AUSSCR alpha	Ausf.Schraube

A.2. Konvexe und nicht konvexe Cluster



(a) konvexes Cluster

(b) nicht konvexes Cluster

Literaturverzeichnis

[Berthold u. a. 2007] Berthold, Michael R. ; Cebron, Nicolas ; Dill, Fabian ; Gabriel, Thomas R. ; Kötter, Tobias ; Meinl, Thorsten ; Ohl, Peter ; Sieb, Christoph ; Thiel, Kilian ; Wiswedel, Bernd: KNIME: The Konstanz Information Miner. In: *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*, Springer, 2007. – ISBN 978-3-540-78239-1

[Cleve und Lämmel 2016] Cleve, Jürgen ; Lämmel, Uwe: *DataMining*. 2. Auflage. DE GRUYTER OLDENBOURG, 2016

[Ertel 2016] Ertel, Wolfgang: *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung*. 4. Auflage. Springer Vieweg, 2016. – 404 S. – ISBN 978-3658135485

[Ester und Sander 2000] Ester, M. ; Sander, J.: *Knowledge Discovery in Databases: Techniken und Anwendungen*. Springer Berlin Heidelberg, 2000. – URL https://books.google.de/books?id=QNat6WM73Q8C. – ISBN 9783540673286

[Ester u. a. 1996] Ester, Martin ; Kriegel, Hans-Peter ; Sander, Jörg ; Xu, Xiaowei: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, AAAI Press, 1996, S. 226–231. – URL www.aaai.org/Papers/KDD/1996/KDD96-037.pdf; abgerufen am 12.12.2016

[Ester und Sander 2013] Ester, Martin ; Sander, Jörg: *Knowledge Discovery in Databases: Techniken und Anwendungen*. 1. Auflage. Springer-Verlag Berlin Heidelberg, 2013

[Falk u. a. 2016] Falk, Dietmar ; Krause, Peter ; Tiedt, Günther: *Metalltechnik Tabellenbuch*. 5. Auflage. westermann, 2016

[Fayyad u. a. 1996] Fayyad, Usama M. ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic: From Data Mining to Knowledge Discovery: An Overview. In: Fayyad, Usama M. (Hrsg.) ; Piatetsky-Shapiro, Gregory (Hrsg.) ; Smyth, Padhraic (Hrsg.) ; Uthurusamy, Ramasamy (Hrsg.): *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA, USA : American Association for Artificial Intelligence, 1996, S. 1–34. –

URL http://dl.acm.org/citation.cfm?id=257938.257942; abgerufen am 08.03.2017. – ISBN 0-262-56097-6

[Herden 2015] Herden, Olaf: Data Mining. In: Kudrass, Thomas (Hrsg.): *Taschenbuch Datenbanken*, Carl Hanser Verlag GmbH & Co. KG, 2015

[Hildebrand u. a. 2008] Hildebrand, K. ; Gebauer, M. ; Hinrichs, H. ; Mielke, M.: *Daten- und Informationsqualität*. 1. Auflage. Vieweeg+Teubner Verlag, 2008

[J.A. und M.A. 1979] J.A., Hartigan ; M.A., Wong: Algorithm AS 136: A K-Means Clustering Algorithm. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28 (1979), Nr. 1, S. 100–108. – URL www.jstor.org/stable/2345630; abgerufen am 12.12.2016

[Kaufman und Rousseeuw 2008] Kaufman, Leonard ; Rousseeuw, Peter J.: *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Inc., 2008. – URL http://onlinelibrary.wiley.com/book/10.1002/9780470316801; abgerufen am 18.03.12. – ISBN 9780470316801

[Kim und Seo 1991] Kim, W. ; Seo, J.: Classifying schematic and data heterogeneity in multidatabase systems. In: *Computer* 24 (1991), Dec, Nr. 12, S. 12–18. – ISSN 0018-9162

[KNIME] KNIME: *KNIME Analytics Platform*. – URL https://www.knime.org

[Macqueen 1967] Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1967, S. 281–297. – URL www-m9.ma.tum.de/foswiki/pub/WS2010/CombOptSem/kMeans.pdf; abgerufen am 12.12.2016

[Manhart 2017] Manhart, Dr. K.: FAQ Machine Learning: Was Sie über Maschinelles Lernen wissen müssen. (2017), Januar. – URL http://www.computerwoche.de/a/was-sie-ueber-maschinelles-lernen-wissen-muessen,3329560; abgerufen am 21.03.2017

[R Core Team 2016] R Core Team: *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing (Veranst.), 2016. – URL https://www.R-project.org

[RapidMiner] RapidMiner: *RapidMiner Studio*. – URL https://rapidminer.com/

[Rousseeuw 1987] Rousseeuw, Peter J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. In: *Journal of Computational and Applied Mathematics* 20 (1987), S. 53 – 65. – URL www.sciencedirect.com/science/article/pii/0377042787901257; abgerufen am 1.12.2016. – ISSN 0377-0427

Glossar

Big Data	Umfangreiche Mengen an semi-strukturierten, sowie unstrukturierten Daten.
Binning	Die Klasseneinteilung.
Centroid	Mittelpunkt bzw. Mittelwert der einem Cluster zugeordneten Objekte.
Cloud	Bereitstellung von IT-Infrastruktur oder Software über das Internet.
Cluster	Gruppe ähnlicher Objekte.
Euklidische Distanz	Der Abstand von zwei Punkten im n-dimensionalen Raum.
K-nearest-neighbour	Die k nächsten Nachbarn eines Objektes.
Konvexe Cluster	Objekte, die auf einer Strecke zweier Objekte liegen, gehören dem gleichen Cluster an.
Medoid	Typischer Vertreter eines Clusters.
Metadaten	Enthalten Informationen über Merkmale anderer Daten, aber nicht die Daten selbst.
Metrische Daten	Daten mit der Ordnungsrelation der reelen Zahlen mit denen gerechnet werden kann.
Nicht konvexe Cluster	Objekte, die auf einer Strecke zweier Objekte liegen, können einem anderen Cluster angehören.
Nominale Daten	Daten ohne Ordnungsrelation mit denen nicht gerechnet werden kann.

Open-Source	Quelloffene Software.
Ordinale Daten	Daten mit Ordnungsrelation (wie < und >) mit denen nicht gerechnet werden kann.
Toolchain	Systematische Sammlung von Werkzeug-Programmen, die für die Erstellung eines Produktes zur Anwendung kommen.
Weka	Softwaretool, das verschiedene Techniken für den KDD Prozess bereitstellt.

Abkürzungsverzeichnis

CRISP	Cross Industry Standard Process.
DBScan	Density-Based Spatial Clustering of Applications with Noise.
FAQ	Frequently Asked Questions.
KDD	Knowledge Discovery in Databases.
KI	Künstliche Intelligenz.
KNIME	Konstanz Information Miner.
Weka	Waikato Environment for Knowledge Analysis.
YALE	Yet Another Learning Environment.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 4. April 2017 Tobias Braack