HAW HAMBURG

BACHELORTHESIS
Daria Zhdanova

Data Cleansing Ansätze zur Verbesserung der Datenqualität am Beispiel fehlerhafter Testdaten für Data Mining

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Daria Zhdanova

Data Cleansing Ansätze zur Verbesserung der Datenqualität am Beispiel fehlerhafter Testdaten für Data Mining

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Wirtschaftsinformatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

[1] Studentin im 9. Semester des Studiengangs Wirtschaftsinformatik
Korrespondenz: [redacted]

Betreuender Prüfer: Prof. Dr. Ulrike Steffens
Zweitgutachter: Prof. Dr. Stefan Sarstedt

Eingereicht am: 02.10.2025

Daria Zhdanova

Thema der Arbeit

Data Cleansing Ansätze zur Verbesserung der Datenqualität am Beispiel fehlerhafter Testdaten für Data Mining

Stitchwort

Data Cleansing, Data Mining, Data Quality

Kurzzusammenfassung

In dieser Arbeit werden verschiedene Ansätze zur Bereinigung von Daten für Mining-Prozesse untersucht und verglichen. Mithilfe von Werkzeugen wie Python-Pipelines, ETL-Prozessen sowie ML-basierten Ansätzen werden fehlerhafte Testdaten aus unterschiedlichen Quellen auf verschiedene Weise bereinigt. Der Schwerpunkt liegt auf der Evaluierung dieser Methoden und deren Einfluss auf die Genauigkeit der Data-Mining-Ergebnisse.

Daria Zhdanova

Title of thesis

Data cleansing approaches for improving data quality based on faulty test data in data mining.

Keywords

Data Cleansing, Data Mining, Data Quality

Abstract

This study examines and compares various approaches to data cleansing for mining processes. Using tools such as Python pipelines, ETL workflows and machine learning-based approaches, faulty test data from multiple sources areas cleansed in different ways. The focus is on evaluating these methods and their influence on the accuracy of the data mining results.

Inhaltsverzeichnis

Abbildungsverzeichnis ..... **vi**

Listings ..... **vii**

Tabellenverzeichnis ..... **viii**

Abkürzungsverzeichnis ..... **ix**

Glossar ..... **x**

1 Einleitung **1**

1.1 Problemstellung und Motivation 1

1.2 Ziele der Bachelorarbeit 1

1.3 Aufbau der Bachelorarbeit 2

2 Hintergrund der Datenbereinigung **3**

2.1 Datenqualitätsprobleme 4

2.1.1 Auswirkungen mangelnder Datenqualität 4

2.1.2 Arten und Ursachen von Datenqualitätsproblemen 5

2.2 Data Cleansing 9

2.2.1 Datenbereinigungszyklus 9

2.2.2 Methoden und Techniken zur Datenbereinigung 11

2.2.3 Werkzeuge zur Datenbereinigung 15

3 Experimentelle Analyse **19**

3.1 Methodische Basis: Testdaten, Tools und Vorgehen 19

3.1.1 Testdatengenerierung 19

3.1.2 Toolauswahl 24

3.1.3 Aufbau des allgemeinen Bereinigungsalgorithmus 25

3.2 Technische Umsetzung der Bereinigungsverfahren 27

3.2.1 Entwurf mit Python 27
3.2.2 Entwurf mit Pentaho DI 33
3.2.3 Entwurf mit Dataiku DSS 38
3.3 Evaluierung und Vergleich der Ergebnisse 42
4 Fazit und Diskussion 49
4.1 Schlussfolgerungen und Ausblick 49
4.2 Limitationen des Experiments 52
4.3 Diskussion 53
Literaturverzeichnis 55
A Anhang 1 60
B Anhang 2 61
C Anhang 3 62
D Anhang 4 63

Abbildungsverzeichnis

Abbildung 1. Ablauf eines Data-Mining-Prozesses. 3
Abbildung 2. Datenqualitätsmängel nach Wang und Strong (1996)..... 6
Abbildung 3. Iterativer Datenbereinigungszyklus nach Ilyas (2016)..... 10
Abbildung 4: Zentrale Datenbereinigungsmethoden nach Dimensionen..... 12
Abbildung 5: ER-Diagramm des Testdatensatzes..... 21
Abbildung 6: Boxplot der numerischen Felder vor der Ausreißerbereinigung. 32
Abbildung 7: Boxplot der numerischen Felder nach der Ausreißerbereinigung..... 32
Abbildung 8: Ablauf des Datenbereinigungs-Jobs in Pentaho..... 34
Abbildung 9: Erste Transformationsstufe: Standardisierung und Deduplizierung. 34
Abbildung 10: Zweite Transformationsstufe: Ausreißerbereinigung. 35
Abbildung 11: Dritte Transformationsstufe: Behandlung fehlender Werte. 37
Abbildung 12: Beispiel für eine statistische Datenübersicht in Dataiku. 39
Abbildung 13: Formatierung von Telefonnummern mittels KI-Agenten in Dataiku..... 39
Abbildung 14: Von Dataiku bereitgestellte Ausreißerbereinigung. 40
Abbildung 15: ML-Modelle zum Umgang mit fehlenden Werten in Dataiku im Vergleich. . 41
Abbildung 16: Bewertungsmatrix der untersuchten Datenbereinigungstools..... 42
Abbildung 17: Bearbeitungszeit im Vergleich (in Sekunden). 45
Abbildung 18: Durchschnittliche CPU-Auslastung pro Ausführung. 45
Abbildung 19: Qualitätsvergleich nach Bereinigungsschritten. 46
Abbildung 20: Qualitätsvergleich von Imputationstechniken anhand der R^2-Werte. 47

Listings

Listing 1: Struktur der Ausführungspipeline des Frameworks. 28
Listing 2: Funktion zur Erstellung von Datenqualitätsprotokollen. 29
Listing 3: Pipeline zur Standardisierung von Telefonnummern. 30
Listing 4: Ausreißerbereinigung anhand des IQR-Verfahrens. 31
Listing 5: Auffüllen der fehlenden Werte in „Customer_Tenure“ mittels KNN-Methode. 33
Listing 6: Ausreißerbereinigung per SQL-Anweisung im Pentaho-Workflow. 35

Tabellenverzeichnis

Tabelle 1: Typische Datenqualitätsprobleme und ihre Ursachen. 8

Tabelle 2: Übersicht der Testdatenattribute mit gezielten Inkonsistenzen. 22

Tabelle 3: SWOT-Analyse verwendeter Bereinigungstechniken. 51

Abkürzungsverzeichnis

SQL	Structured Query Language
ETL	Extract, Transform, Load
TDQM	Total Data Quality Management
DWQ	The Datawarehouse Quality Methodology
TIQM	Totalinformation Quality Management
BI	Business Intelligence

Glossar

Data Preprocessing	Die Datenvorverarbeitung, ein Bestandteil der Datenaufbereitung, bezeichnet die Verarbeitung von Rohdaten, um sie für andere Datenverarbeitungsverfahren vorzubereiten.
Data Cleansing	Data Cleansing stellt einen systematischen Prozess zum Entfernen und Korrigieren von Datenfehlern dar.
Data Mining	Unter Data-Mining versteht man die gezielte Analyse umfangreicher Datenmengen zur Erkennung neuer Zusammenhänge und Trends.
Dirty Data	Daten, die fehlerhaft, unvollständig, inkonsistent, veraltet oder anderweitig mangelhaft sind.
Data Quality	Alle Maßnahmen, die dazu dienen, die Qualität von Daten zu verbessern und dauerhaft einen hohen Standard des Datenbestandes sicherzustellen.
Data Profiling	Systematische Analyse von Daten zur Identifizierung von Fehlern, Inkonsistenzen und Unvollständigkeiten.
Ad-hoc-Analyse	Spontane Datenanalyse zur schnellen Beantwortung von Fragen.

1 Einleitung

1.1 Problemstellung und Motivation

Mit der wachsenden Anzahl und Vielfalt an Informationsquellen und -typen steigt die Bedeutung der Datenpräzision für alle darauf basierenden Prozesse erheblich. Probleme wie strukturelle oder inhaltliche Fehler, doppelte Einträge oder Inkonsistenzen erfordern einen geeigneten Bereinigungsprozess. Die Wahl des richtigen Ansatzes zur Bereinigung der Rohdaten ist für die Analyse und die Gewinnung neuer Informationen von erheblicher Bedeutung.

Ein illustratives Beispiel dafür fand im Oktober 2020 in Großbritannien statt: Inkompatible Datenformate und die Überschreitung der maximalen Zeilenanzahl in Excel-Dateien führten dazu, dass mehr als 15.000 COVID-19-Fälle nicht in die offizielle Statistik aufgenommen wurden [4]. Dies führte zur falschen Interpretation der Fakten, was in der Folge auch weitere relevante Konsequenzen haben könnte.

1.2 Ziele der Bachelorarbeit

In dieser Studie werden drei fundamentale Methoden der Datenbereinigung im Hinblick auf Data-Mining-Prozesse verglichen. Hierzu werden in einem praktischen Experiment spezielle Instrumente zur Bereinigung desselben Testdatensatzes eingesetzt, der eine Vielzahl typischer qualitativer Probleme umfasst. Die Zielsetzung ist es, die Funktionalität, Benutzerfreundlichkeit, Systemauslastung und weitere Merkmale der Datenbereinigungsprozesse durch theoretische Prinzipien zu bewerten. Anhand der Ergebnisse dieser Bewertung werden Schlussfolgerungen zur Effizienz und zu den Unterschieden der Verfahren gezogen.

1.3 Aufbau der Bachelorarbeit

Die Arbeit ist in drei Hauptabschnitte gegliedert. Das zweite Kapitel umfasst die theoretischen Grundlagen der Datenbereinigung. Hier werden der Stellenwert und die Besonderheit dieses Prozesses beschrieben, ebenso die Hauptarten von Datenverschmutzung und die Methoden und Instrumente zu ihrer Beseitigung. Kapitel 3 befasst sich mit der Dokumentation des praktischen Experiments. Zu Beginn werden die methodologischen Aspekte des getesteten Musters, des Arbeitsalgorithmus und der ausgewählten Reinigungstechnologien in Form der verwendeten Werkzeuge festgelegt. Der größte Teil dieses Abschnitts besteht aus der Dokumentation der technischen Umsetzung und der durchgeführten Analyse. Im vierten Kapitel werden die Forschungsergebnisse in Form von Schlussfolgerungen zusammengeführt. Die letzten Aspekte behandeln die aufgetretenen Einschränkungen, deren Einfluss auf die Arbeitsergebnisse und die Aussichten für weitere Untersuchungen.

2 Hintergrund der Datenbereinigung

Die Fähigkeit, aus umfangreichen Datenmengen verwertbares Wissen zu extrahieren, gewinnt mit deren stetig wachsender Verfügbarkeit an Bedeutung. Hierbei wird das sogenannte Knowledge Discovery in Databases (KDD) verwendet – die Grundlage für datengetriebene Entscheidungen. Data Mining stellt einen entscheidenden Teilprozess dar, bei dem in den Daten hilfreiche Muster und Trends ausgemacht werden [29, S. 1].

Für diesen Prozess ist die Qualität der Basisdaten entscheidend. Daten, die unvollständig oder inkonsistent sind, verursachen nicht nur falsche Analysen, sondern bergen auch ein erhebliches Risiko für Fehlentscheidungen. Data Mining ist nicht als isolierter Analyseschritt zu verstehen, sondern gehört zu einem mehrstufigen Prozess (siehe Abbildung 1) [5].

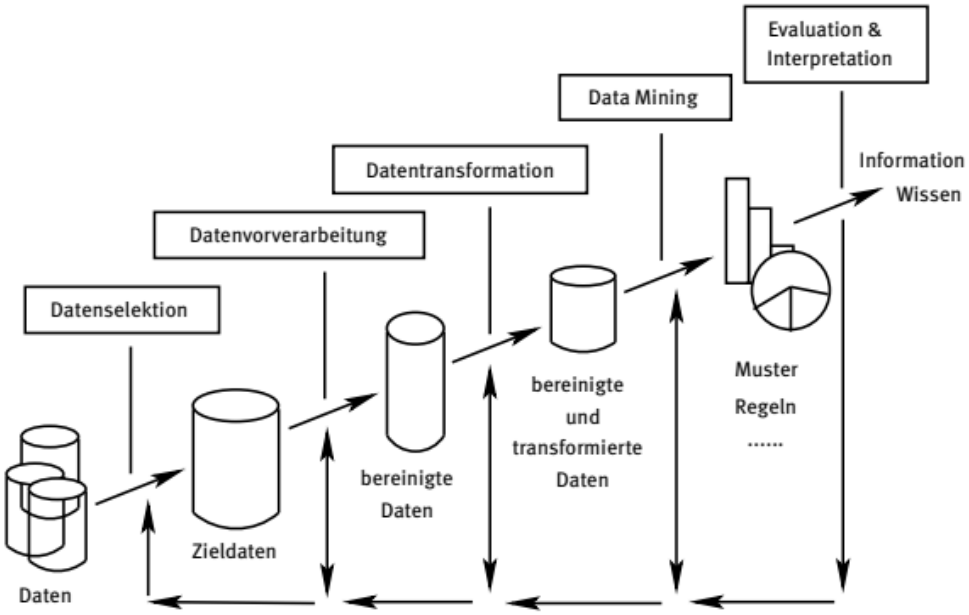


Abbildung 1: Ablauf eines Data-Mining-Prozesses [5].

Die Datenvorverarbeitung ist ein wesentlicher Bestandteil dieses Gesamtprozesses. Sie umfasst das Bereinigen, Transformieren, Integrieren und Reduzieren von Datensätzen und gewährleistet eine stabile, konsistente und qualitativ hochwertige Basis für alle weiteren Schritte [12]. In den kommenden Kapiteln wird diese Phase zunächst aus methodischer Sicht und danach praxisbezogen anhand typischer Verfahren und Instrumente untersucht.

2.1 Datenqualitätsprobleme

Es ist erforderlich, spezifische Problemfälle unzureichender Datenqualität zu identifizieren, um Bereinigungsmaßnahmen richtig und effektiv zu gestalten. Dabei kommen oft die Begriffe „Dirty Data“ und „Noisy Data“ sowie eine hierarchische Einteilung zum Einsatz. Dieses Kapitel veranschaulicht Auswirkungen, Arten und Ursachen von Datenqualitätsproblemen.

2.1.1 Auswirkungen mangelnder Datenqualität

Wie schon angesprochen sind vertrauenswürdige Daten das unverzichtbare Fundament für darauf basierende Entscheidungen und automatisierte Prozesse. Sind jedoch strukturelle Mängel, Fehler oder Lücken in Daten vorhanden, bezeichnet man sie als „dirty data“ – ungenaue, fehlende Angaben und nicht standardisierte Darstellungen derselben Daten, die bei hoher Konzentration in Datenbeständen zu einer instabilen Datenverwaltung führen können [45]. Im Gegensatz dazu stellt „clean data“ eine konsistente und zuverlässige Grundlage dar, die – vor allem im wirtschaftlichen Bereich – etwa zur Effizienzsteigerung, zur Verbesserung der Kundenansprache und zur Einhaltung von Vorschriften beiträgt [2].

Ein Praxisbeispiel zeigt diese Problematik: Bevor ein Einzelhandelsanbieter eine Entscheidung über die Durchführung einer Marketingkampagne trifft, möchte er die Verkaufsdaten für verschiedene Produktkategorien analysieren. Die gewünschten Informationen sind jedoch schwer zugänglich oder unvollständig, da verschiedene Daten aus mehreren Quellen fehlerhaft oder falsch integriert wurden. So kam es in der Phase des Data Mining zu Fehlern bei den Prognosen und zu Entscheidungen, die nicht der Realität entsprachen.

Eine Form von „dirty data“ sind fehlerhafte Daten, auch „noisy data“ genannt. Dabei handelt es sich oft um zufällige Fehler in Variablen, die deren tatsächlichen Wert verändern [45]. Ein

typisches Beispiel wären unstrukturierte Freitextkommentare in Datensätzen, bei denen Tippfehler oder einzeln stehende Begriffe fälschlicherweise als relevant erkannt werden. Dies kann die Verarbeitung reduzieren und die Aussagekraft datenbasierter Verfahren erschweren.

Die Anforderungen an Daten unterscheiden sich je nach Fachbereich ebenfalls erheblich. In der Lagerverwaltung sind bestimmte Informationen beispielsweise nicht erforderlich, während sie für das Marketing oder das Controlling unerlässlich sind. Der Informationswert wird zudem maßgeblich von Faktoren wie Aktualität, Nachvollziehbarkeit und struktureller Einheitlichkeit beeinflusst.

In der praktischen Anwendung kann es sich allerdings als nicht wirtschaftlich herausstellen, sämtliche bestehende Datenbestände zu bereinigen. Wie Haug et al. (2011) anhand einer Fallstudie zeigen, entstanden bei einem Ersatzteilhersteller erhebliche Zusatzkosten durch fehlerhafte und inkonsistente Einträge in Datenbanken. Die vollständige Korrektur aller Mängel hat unverhältnismäßigen Aufwand verursacht. Stattdessen wurde ein optimales Niveau der Datenqualität angestrebt, auf dem sich Aufwand und Nutzen in einem angemessenen Verhältnis befinden. Dies zeigt, dass für datenbasierte Entscheidungen nicht Perfektion, sondern vielmehr ein an den Verwendungszweck angepasster Qualitätsstandard entscheidend ist [13].

2.1.2 Arten und Ursachen von Datenqualitätsproblemen

Abhängig von der konzeptionellen Sichtweise kann man die Datenqualität in verschiedene Dimensionen aufteilen. Eine der einflussreichsten Studien in diesem Bereich ist die von Wang und Strong (1996), die als theoretische Grundlage für das Management der Datenqualität dient. Sie legen in ihrer Arbeit vier grundlegende Kategorien innerhalb einer hierarchischen Struktur fest: inhärente, kontextbezogene, darstellungsbezogene und systemgestützte Datenqualität. Jede dieser Kategorien veranschaulicht bestimmte Aspekte der Datenqualität, die als nicht erfüllte Anforderungen in konkrete Problempunkte umgewandelt werden können (Abbildung 2) [50].

Die intrinsische Dimension umfasst die inhaltliche Qualität der Daten, welche durch Faktoren wie Genauigkeit, Glaubwürdigkeit und Objektivität evaluiert wird [50].

Die kontextuelle (oder extrinsische) Dimension betrachtet hingegen die Datenqualität in Bezug auf eine spezifische Aufgabe. Dabei sind die Relevanz, der Aktualitätsgrad, die Vollständigkeit und der Nutzen der Daten entscheidend [50].

Die letzten beiden Dimensionen unterstreichen die Bedeutung datengestützter Systeme, die Daten konsistent, sicher, zugänglich und verständlich bereitstellen.

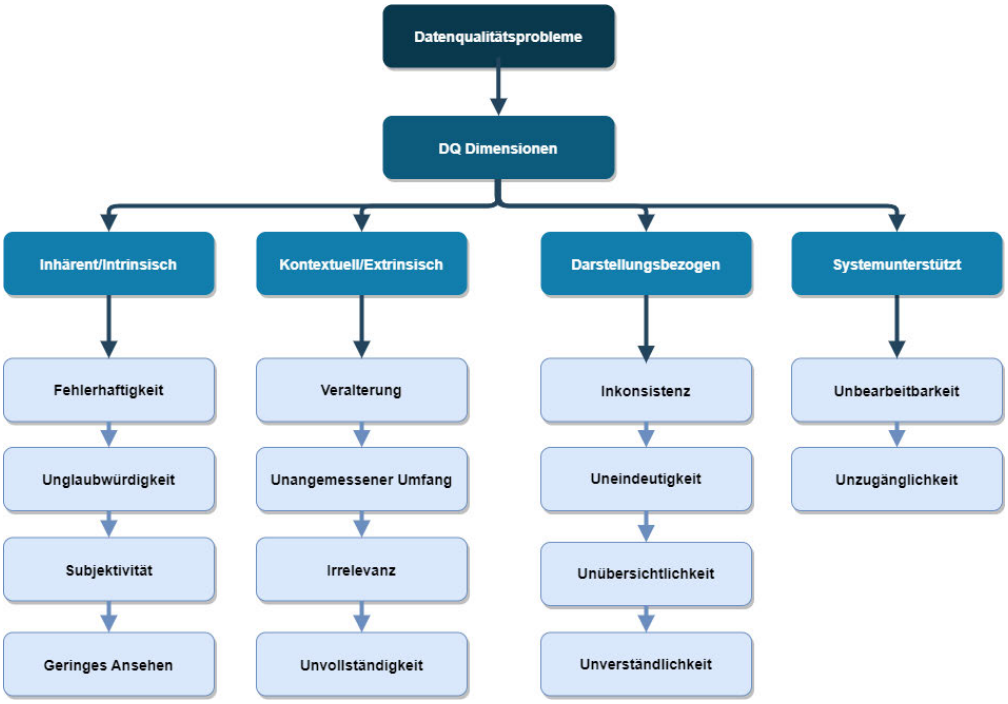


Abbildung 2: Datenqualitätsmängel nach Wang und Strong (1996) [50].

Diese Kategorisierung stellt eine konzeptionelle Grundlage dar und beschreibt alle vier Dimensionen umfassend. Im technischen Verarbeitungsprozess werden diese Fälle jedoch oft nicht isoliert, sondern als zusammenwirkend betrachtet. Dadurch lassen sich Probleme mit „verschmutzten" Daten genauer in folgende vier Gruppen unterteilen: Fehlerhaftigkeit, Inkonsistenz, Inkompatibilität und Unvollständigkeit.

Unvollständigkeit:

Unvollständigkeit bedeutet, dass Daten fehlen, die erforderlich sind, um reale Objekte oder Prozesse vollständig und korrekt abzubilden. Es ist entscheidend, ob ein Wert tatsächlich nicht existiert oder nicht erfasst wurde. In relationalen Datenbanken werden fehlende Informationen

häufig mit Nullwerten dargestellt. Je nach Anwendungszusammenhang können sie bewusst hingenommen, durch Normwerte ersetzt oder als Ausschlusskriterium in Auswertungen verwendet werden. Wenn die systematische Verteilung des Fehlens nicht verständlich ist, wird dieses Problem gravierender.

Fehlerhaftigkeit:

Fehlerhafte Daten äußern sich in Informationen, die entweder falsch, unzuverlässig oder nicht mit der Realität vereinbar sind. Zu möglichen Ursachen gehören unter anderem ungenaue Messungen, manuelle Eingabefehler und das Fehlen von Integritätsprüfungen. So kann beispielsweise durch einen Tippfehler statt „1987" der Wert „1897" in der Spalte „Geburtsjahr" eingetragen werden. Ein weiteres Problem sind doppelte Datensätze (Dubletten). Sie können entstehen, wenn dieselbe Entität durch eine Übernahme aus mehreren Systemen oder durch mehrfache Erfassung – doppelt oder leicht abweichend – vorkommt. Beispiele dafür sind die Namenseinträge „Max Müller" und „Max Mueller". Außerdem gehören dazu uneinheitlich verwendete Abkürzungen oder fehlerhafte Umrechnungen in Datensätzen.

Inkonsistenz:

Daten weisen Inkonsistenzen auf, wenn semantische Regeln, die als Integritätsbedingungen bekannt sind, verletzt werden. Diese Regeln legen fest, welche Werte in einem Datenbestand zulässig sind und wie sie miteinander in Beziehung stehen müssen. In relationalen Datenbanken wird dabei zwischen intrarelationalen (z. B. „Das Alter zwischen 0 und 120") und interrelationalen Bedingungen (z. B. „Das Eintrittsdatum eines Mitarbeiters vor dem Austrittsdatum") unterschieden [2, S. 7]. Verschiedene Bezeichnungen für ein und dasselbe Objekt oder die gemischten, zusammengesetzten Attribute sind ebenfalls typische Beispiele für Inkonsistenzen.

Inkompatibilität:

Wenn ihre Formate oder Strukturen den Anforderungen der Zielsysteme nicht entsprechen, gelten solche Daten als inkompatibel. Typische Beispiele sind uneinheitliche Datumsformate (z. B. das US-Format „MM/DD/YYYY" im Vergleich zum EU-Format „DD/MM/YYYY"), falsch organisierte Strukturen (z. B. Eingaben in Zeilen statt in Spalten), fehlerhafte Zeichencodierungen (z. B. Umlaute bei der Konvertierung von UTF-8/ASCII) oder HTML-Tags, die in Textfeldern eingebettet sind. Heterogene Quellsysteme – wie ERP-Anwendungen,

Webformulare oder Excel-Dateien – ohne abgestimmte Formatvorgaben, Zeichencodierung oder übergreifendes Datenmodell zu integrieren, ist häufig der Grund für solche Inkompatibilitäten.

Tabelle 1 veranschaulicht diese Kategorien anhand konkreter Probleme und ihrer Ursachen.

Tabelle 1: Typische Datenqualitätsprobleme und ihre Ursachen.

Datenqualitätsprobleme		**Beschreibung**	**Beispiel**	**Mögliche Ursache**
Kategorie	***Problemart***			
Unvollständigkeit	Fehlende Werte	Leere oder nicht ausgefüllte Felder	Telefonnummer fehlt	Systemfehler, manuelle Auslassung
Fehlerhaftigkeit	Dubletten	Mehrfacheinträge desselben Objekts	Name: „Max Müller“ und „Maximilian Müller“	Fehlende Abgleichregeln
	Fehlerhafte Werte	Ungültige oder unplausible Daten	Geburtsdatum: „31.02.1990“, „01.01.2090“	Tippfehler, fehlende Plausibilitätsprüfungen
Inkompatibilität	Formatfehler	Uneinheitliche Darstellungen	Rabatt: „20 %“ und „0,2“	Unterschiedliche Eingabemuster
Inkonsistenz	Logische Widersprüche	Widersprüchliche Angaben innerhalb eines Datenkontexts	Alter (in Jahren): „18“ und Mitgliedschaftsjahre: „20“	Fehlende Prüfung von Kohärenzeinschränkungen

Die Herausforderungen nehmen zu, wenn Daten aus unterschiedlichen Quellen kombiniert werden: Die unterschiedlichen Strukturen und Bezeichnungen von Daten können leicht die Schema-Konflikte verursachen. Probleme können schon auf der Inhaltsebene entstehen, zum Beispiel durch unterschiedliche Datentypen, Aggregationslogiken oder zeitliche Referenzen. Dies erschwert die eindeutige Zuordnung von Datensätzen, die dieselben Entitäten in unterschiedlichen Geschäftskontexten repräsentieren, wie etwa doppelte Kundeneinträge aus der Sicht von Marketing und von Vertrieb [28, S. 2-5].

Wenn Forscher und Praktiker über die möglichen Arten von Datenproblemen informiert sind, können sie effektivere Strategien zur Datenverwaltung entwickeln. Unscharfe Daten können durch ein durchdachtes Datenbankdesign und gezielte Einschränkungen schon während der Integrationsphase des Informationssystems verhindert werden. Dies ist entscheidend für die Verbesserung der Steuerung von Mining- und Analyseprozessen.

2.2 Data Cleansing

Data Cleansing ist der Prozess, der strukturierte technische und semantische Aktionen umfasst, um Probleme mit der Datenqualität zu beheben. In der Regel besteht dieses Vorgehen aus zwei Phasen: dem Erkennen und dem Beheben von Datenfehlern [20]. In dieser Arbeit werden sie sowohl aus theoretischer als auch aus praktischer Sicht betrachtet.

2.2.1 Datenbereinigungszyklus

Laut Ilyas (2016) werden zahlreiche Unstimmigkeiten erst bei Auswertungen, Berichten oder Analysen deutlich [19]. Das heißt, die Inkonsistenzen werden von Business-Analysten nicht in den Quelldaten, sondern in den darauf aufbauenden Sichten der Datenverarbeitungspipeline erkannt. Dies hebt ein bedeutendes Problem hervor: die Entkopplung von Fehlern und deren Entdeckung in Raum und Zeit [19].

Aufgrund dieser Verzögerungen wird eine zeitnahe Fehlerbehebung erschwert. Stattdessen definiert Ilyas (2016) einen iterativen Bereinigungszyklus für die Datenverarbeitung. Ein Prozess, der als „Clean-Evaluate-Loop“ (engl.) bekannt ist, bringt dieses Vorgehen zusammen mit der aktiven Nutzerbeteiligung und den Konzepten der Datenprovenienz [19] (Abbildung 3).

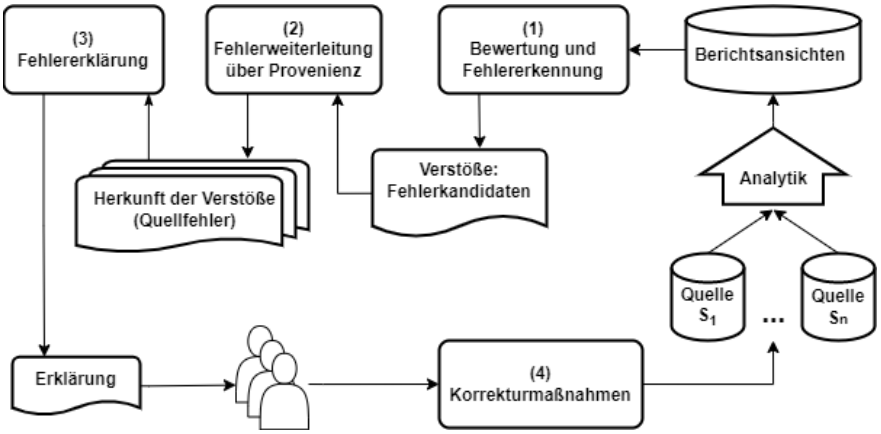


Abbildung 3: Iterativer Datenbereinigungszyklus nach Ilyas (2016) [19].

Der Zyklus wird als ein Kreislauf zur Evaluation und Bereinigung dargestellt, in dem potenzielle Fehler erkannt, mögliche Erklärungen dafür formuliert und in konkrete Bereinigungsaktionen umgesetzt werden. Er besteht aus vier Hauptschritten:

1. Bewertung und Fehlererkennung

Fehler werden in erster Linie auf der Ebene der Berichterstattung erkannt. Es kann unter anderem durch automatisierte Prüfroutinen (wie zur Identifizierung von Integritätsverletzungen), statistische Ausreißererkennung oder manuelle Analysen durch Fachanwender umgesetzt werden. Zudem erfolgt eine Überprüfung der Auswirkungen von Bereinigungsmaßnahmen, die zuvor durchgeführt wurden, beispielsweise durch A/B-Tests oder Vergleichsanalyse.

2. Fehlerweiterleitung durch Provenienz

In der nächsten Phase werden die konkreten problematischen Quellwerte bestimmt. Je mehr die Datenverarbeitung verkompliziert wird, etwa durch Aggregationen, desto anspruchsvoller gestaltet sich ihre Rückverfolgbarkeit. Um die fehlerverursachenden Daten so präzise wie möglich zu identifizieren, ist es entscheidend, Beweise für mehrere Regelverstöße zu sammeln. Dies erfolgt mittels automatischer Traceback- oder Lineage-Analysen (z. B. Scorpion, DBRx).

3. Fehlererklärung

Nach der Rückverfolgung der fehlerhaften Berichtswerte werden Merkmale in den Quelldaten ausgemacht, die direkt mit den festgestellten Regelverstößen verknüpft sind. Das Ziel ist es, dass aus den betroffenen Datenbereichen klare und nachvollziehbare Erklärungen entstehen,

die als Basis für Korrekturmaßnahmen dienen. Laut Ilyas können dafür sinnvolle Views auf die Datenquellen aufgebaut werden, die die propagierten Fehler darstellen.

4. Bereinigungsmaßnahmen

Der Zyklus wird beendet, sobald Fachleute spezifische Korrekturmaßnahmen basierend auf den identifizierten Fehlerursachen festlegen. Dazu gehören unter anderem Transformationen, die zielgerichtete Anpassung spezifischer fehlerhafter Werte und das Entfernen von redundanten sowie nicht vertrauenswürdigen Datenquellen. Die Berichte werden nach Umsetzung der Anpassungen analysiert, um die Auswirkungen der Änderungen zu bewerten. Wenn die Probleme bestehen bleiben, wird ein neuer Zyklus gestartet.

2.2.2 Methoden und Techniken zur Datenbereinigung

Ilyas und Chu (2019) haben in ihrer Studie beschrieben, dass die methodischen Konzepte des Data Cleansings durch zwei Dimensionen strukturiert werden können: den Ansatz, den man verfolgt, und das spezifische Ziel. Der qualitative Ansatz betrachtet strukturelle und regelbasierte Aspekte der Daten, während der quantitative Ansatz sich auf statistische Abweichungen fokussiert. Zudem wird zwischen Methoden zur Erkennung und zur Behebung von Fehlern unterschieden. Anhand dieser Dimensionen lassen sich sechs zentrale Aufgaben unterscheiden (Abbildung 4) [20, S. 4-10].

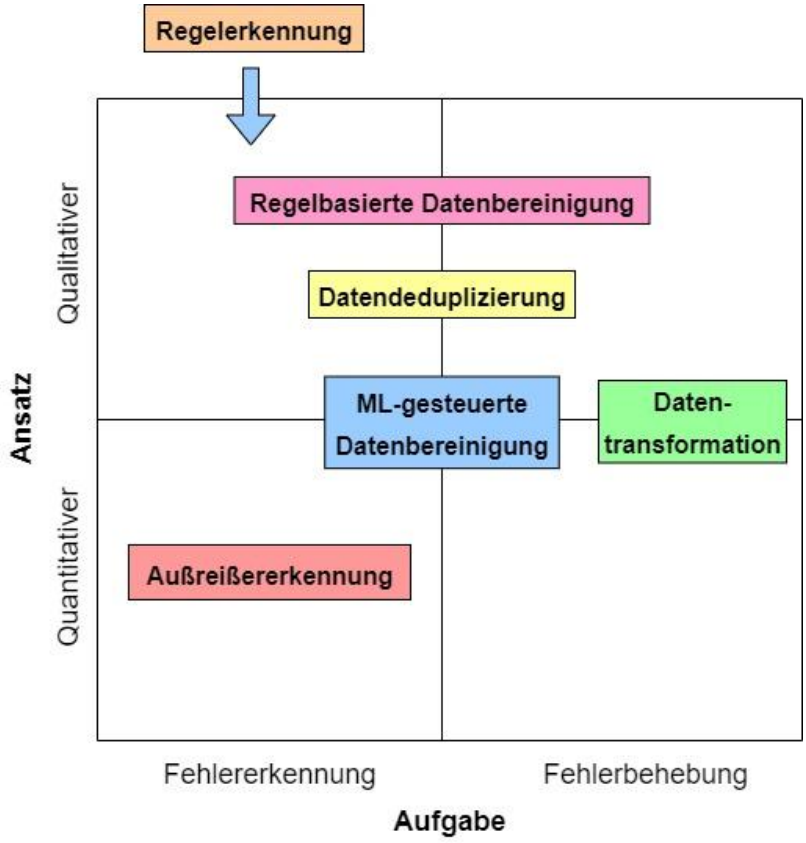


Abbildung 4: Zentrale Datenbereinigungsmethoden nach Dimensionen [20].

Der qualitative methodische Ansatz der ***regelbasierten Datenbereinigung*** dient dazu, Inkonsistenzen zu erkennen und diese basierend auf definierten Datenregeln zu beheben. Mit diesen Regeln wird das semantische Wissen über die Datenstruktur dargestellt. Das beinhaltet funktionale Abhängigkeiten zwischen den Attributen sowie den Ausschluss bestimmter Wertkombinationen. Wenn Grundsätze verletzt werden, versucht das System, einen konsistenten Datensatz durch die Anwendung der folgenden Verfahren wiederherzustellen:

- **Optimierungsbasierte Methoden:** Um die global optimale Lösung mit minimalem Gesamtaufwand zu ermitteln, wird jede mögliche Korrektur mathematisch untersucht.
- **Heuristische Verfahren:** In jedem Schritt wählen sequenzielle Algorithmen die Entscheidung, die lokal am besten ist, ohne den gesamten Lösungsraum durchsuchen zu müssen. Erfüllen mehrere Regeln gleichzeitig die Bedingungen, werden sie als logische Nebenbedingungen (Constraints) zusammengefasst, die im Voraus definiert sind.

Bei der Datendeduplizierung handelt es sich um ein quantitatives Verfahren, dessen Ziel es ist, identische Einträge zu erkennen, die dieselbe reale Entität repräsentieren. Sowohl Schritte zum semantischen Verständnis als auch zur aktiven Anpassung von Daten sind Teil dieses Prozesses:

- **Ähnlichkeitsberechnung:** Die Datensätze werden mittels Eins-zu-eins-Vergleichen oder verschiedener Differenzmessverfahren untersucht. Die Levenshtein-Distanz wird verwendet, um Zeichenfolgen zu vergleichen, während die Jaccard-Ähnlichkeit zur Analyse von Überschneidungen zwischen Wertemengen dient.
- **Duplikaterkennung:** Anhand der berechneten Ähnlichkeitswerte und festgelegten Schwellenwerte wird entschieden, welche Datensätze doppelt vorhanden sind.
- **Konsolidierung:** Es wird entweder der Attributwert mit dem höchsten Vertrauensniveau übernommen oder es werden mehrere Daten zu einem konsistenten Eintrag zusammengeführt.

Die ML-gesteuerte Datenbereinigung wird insbesondere dann eingesetzt, wenn starre oder unvollständige Regeln zur Fehlererkennung nicht ausreichen. Durch die Kombination aus statistischen Zusammenhängen, datengetriebenem Lernen und qualitativen Merkmalen lassen sich drei grundlegende Phasen einer mustergeprägten Datenbereinigung unterscheiden:

- **Annotation:** Markierung korrekter und fehlerhafter Werte im Trainingsdatensatz.
- **Modelltraining:** Die Algorithmen des überwachten Lernens, wie etwa Entscheidungsbäume, neuronale Netze oder Support-Vector-Machines, werden eingesetzt, um die charakteristischen Merkmalsmuster von fehlerfreien und fehlerhaften Einträgen zu identifizieren und neue Datenpunkte automatisiert auf ihre Richtigkeit zu prüfen.
- **Imputation und Korrektur:** Die Anwendung von regressionsbasierten, probabilistischen oder instanzbasierten Methoden – etwa Multiple Imputation, Bayes'sche Netze oder k-Nearest Neighbors (k-NN) –, um fehlende oder abweichende Werte zu schätzen und zu ersetzen, indem man statistische Zusammenhänge im Datensatz nutzt.

Die Datentransformation richtet ihren Fokus vor allem auf die Behebung inkonsistenter Datenformate. Sie ist von erheblicher Bedeutung, wenn die Daten inhaltlich korrekt sind, jedoch nicht einheitlich standardisiert aufbereitet sind – wie zum Beispiel bei unterschiedlich erfassten

Telefonnummern oder Adressinformationen. Dabei werden strukturelle Eigenschaften und Muster, die sich im Datensatz wiederholen, analysiert. In der Regel werden dabei diese Techniken angewendet:

- **Feste Umwandlungsregel:** Wiederkehrende Muster werden, beispielsweise durch reguläre Ausdrücke, identifiziert und entsprechend einer festgelegten Norm konvertiert.
- **Beispielgestützte Transformation:** Automatisierungsregeln lassen sich ableiten, indem individuelle Korrekturfälle analysiert und verallgemeinert werden, sodass ähnliche Abweichungen im Datenformat normiert bereinigt werden können.

Die Identifizierung von Ausreißern basiert auf der statistischen Untersuchung numerischer Merkmale. Die Resultate können als quantitative Indikatoren für den Umgang mit außergewöhnlichen Eingabewerten verwendet werden, die erheblich vom Durchschnitt des Datensatzes abweichen – insbesondere von sehr hohen oder sehr niedrigen Zahlen.

- **Verteilungsbasierte Verfahren:** Beispiele hierfür sind Z-Score und Interquartilsbereich (IQR), bei denen Ausreißer durch Abweichungen vom Mittelwert beziehungsweise außerhalb typischer Quartilsgrenzen identifiziert werden.
- **Dichtebasierte Methoden:** Es wird untersucht, wie viele ähnliche Datenpunkte sich in der Nähe eines bestimmten Wertes befinden. Ein Wert wird als Ausreißer betrachtet, wenn seine lokale Dichte im Vergleich zu anderen Bereichen signifikant geringer ist.
- **Distanzbasierte Methoden**: Basierend auf der Entfernung eines Werts zu seinen nächsten Nachbarn werden Punkte, die ungewöhnlich weit entfernt liegen, als potenzielle Ausreißer identifiziert.

Die Regelerkennung liegt außerhalb der eigentlichen Methodendimensionen und bildet die Grundlage für eine spätere regelbasierte Bereinigung, indem sie aus dem Eingabedatensatz Regeln zur Qualitätssicherung ableitet. Es kommen zwei Verfahren zur Anwendung:

- **Identifikation funktionaler Abhängigkeiten:** Es wird analysiert, ob ein Attributswert einen anderen unabhängig vom Kontext eindeutig bestimmt – etwa, wenn eine Produktnummer im gesamten Datensatz immer mit dem gleichen Preis verknüpft ist.

- **Entdeckung bedingter Regeln:** Es werden Verbindungen herausgefunden, die nur unter bestimmten Bedingungen gelten – zum Beispiel: Die Mehrwertsteuer beträgt dann 7 %, wenn die Kategorie „Buch“ lautet.

Die gezeigten Aufgaben stellen die methodische Grundlage für die praktische Datenbereinigung dar. Je nach Anwendungsart setzen spezialisierte Werkzeuge dabei unterschiedliche Abläufe und Schwerpunkte ein, die im folgenden Unterkapitel näher erläutert werden.

2.2.3 Werkzeuge zur Datenbereinigung

Mit der Komplexität der Datenstruktur, dem Wachstum des Datenvolumens und den unterschiedlichen Verarbeitungsanforderungen ist es entscheidend, passende Bereinigungstools auszuwählen. Wie nützlich und effizient diese Werkzeuge sind, hängt stark davon ab, wie sie Fehler erkennen und korrigieren, sowie davon, wie automatisiert sie sind und welche Benutzerinteraktion möglich ist. Basierend auf diesen Merkmalen und einigen Studien unterscheidet man vier Kategorien von Werkzeugen zur Datenprofilierung und -bereinigung [30, 25, 18]:

Tabellenkalkulationen

Bei der direkten Dateiverarbeitung arbeitet die zuständige Person über eine Tabellenkalkulationsoberfläche und nutzt grundlegende Funktionen zur Datenbearbeitung. Anwendungen wie Microsoft Excel oder Google Sheets haben hierfür Standardfunktionen wie Filter, Formatierungen, Formeln und Aggregationen, die eine regelbasierte oder manuelle Vorgehensweise erlauben [10, 31]. Selbst Anwender ohne umfassende technische Kenntnisse sind dadurch in der Lage, Datenprobleme zu erkennen und gezielt zu beheben. Sie sind besonders geeignet für explorative Analysen, die eine hohe Bearbeitungsgeschwindigkeit erfordern, sowie für die Verarbeitung von kleinen bis mittelgroßen Datenmengen. Diese Werkzeuge erreichen im Vergleich zu ausgereifteren Verfahren insbesondere bei einer steigenden Komplexität der Daten und aufgrund ihrer begrenzten Wiederverwendbarkeit ihre Grenzen.

Skriptbasierte Tools

Die nächste Kategorie bietet eine hohe Flexibilität bei der Steuerung von Profiling- und Bereinigungsprozessen im Datenmanagement. Vor allem, wenn es darum geht, umfangreiche Datenmengen zu verarbeiten oder automatisierte Analyseprozesse einzubinden, sind

skriptbasierte Anwendungen von großer Bedeutung. Häufig werden dabei Programmiersprachen wie Python oder R verwendet, deren Bibliotheken pandas [36], dplyr oder tidyr es ermöglichen, flexible Datenpipelines zu erstellen und komplexe Bearbeitungsschritte durchzuführen. In SQL-basierten Systemen erlauben Constraints eine regelbasierte Vorausprüfung von Daten, während Stored Procedures Aktionen strukturieren und wiederverwendbar machen. Open-Source-Lösungen haben den Vorteil, dass sie kostenlos sind, eine breite Community-Unterstützung haben, sich gut in bestehende Systeme einfügen lassen und die Versionskontrolle als Grundlage für eine verlässliche Weiterentwicklung ermöglicht. Im Gegensatz dazu erfordert der Einsatz aufgrund des Fehlens grafischer Benutzeroberflächen fundierte Programmierkenntnisse, eine Einarbeitungszeit und eine sorgfältige Dokumentation. Für anspruchsvolle Datenbereinigungsprojekte sind solche Lösungen auf lange Sicht eine stabile und effiziente Grundlage.

ETL-Workflows

Visuelle Plattformen mit Drag-and-Drop-Oberflächen ermöglichen es, den Verarbeitungsprozess schrittweise aus vorgegebenen oder selbst erstellten Funktionseinheiten zu bauen. Dieser Ansatz ist entscheidend, wenn es erforderlich ist, einen komplexen und strukturierten Überblick über umfassende Datenströme zu erhalten, die aus verschiedenen Quellen integriert, transformiert und weitergeleitet werden. Ein detailliertes Metadatenmanagement unterstützt dies, indem es die Verwaltung sowie die Nachverfolgbarkeit der Datenherkunft sicherstellt. Diese Eigenschaften machen workflowbasierte Tools zu einer Wahl im produktiven und kommerziellen Bereich, wo die Datengewinnung durch hohe Differenziertheit und Komplexität geprägt ist. Trotz ihrer exzellenten Skalierbarkeit sind solche Werkzeuge bei häufigen Änderungen der Datenstruktur oder der Verarbeitungsstufen für situative Ad-hoc-Analysen weniger geeignet. Je nach Anbieter und bereitgestellten Funktionen kann es eine intensivere Einarbeitung sowie höhere Lizenz- und Betriebskosten verursachen – vor allem im Vergleich zu tabellarischen Lösungen.

Interaktive ML-Technologien

Moderne Datenbereinigungswerkzeuge nutzen intelligente Algorithmen und maschinelles Lernen. Solche Systeme erledigen Aufgaben wie das Erkennen von Mustern, die prädiktive Entscheidungsfindung und die fortlaufende Optimierung von Prozessen. In Anwendungen wie

„Dataprep by Trifacta“ kommen mittlerweile fortschrittliche KI-Agenten zum Einsatz, die eigenständig Vorschläge zur Datenbereinigung sowie teilweise sogar vollständige Lösungswege entwickeln. Auf diese Weise kann man den Aufwand, der manuell erledigt werden müsste, erheblich minimieren. ETL-Plattformen mit umfangreicher Funktionalität haben oft Lösungen, die speziell für die Verarbeitung von massiven Datenmengen entwickelt wurden. Häufig wird die Nutzung jedoch cloudbasiert angeboten, was die erforderliche technologische Infrastruktur voraussetzt und dadurch möglicherweise hohe Betriebs- und Lizenzkosten verursachen kann. Außerdem ist es wichtig, dass diese Systeme sorgfältig die datenschutzrechtlichen und Compliance-bezogenen Vorgaben beachten. Diese Systeme werden zudem meist als Black-Box-Lösungen betrachtet, weil ihre Entscheidungslogik für die Nutzer nicht vollständig verständlich ist. Dies schränkt eine vollständige Automatisierung ohne fachliche Kontrolle ein – auch wenn die Genauigkeit der Ergebnisse durch die lernbasierte Optimierung kontinuierlich steigt.

Tabelle im Anhang A gibt einen umfassenden Überblick über die vier genannten Kategorien, einschließlich typischer Merkmale, Vor- und Nachteile und konkreter Beispiele.

In der Praxis wird immer deutlicher, dass sich viele moderne Datenbereinigungstools nicht streng in eine dieser Kategorien einordnen lassen, da sie kombinierte Ansätze verwenden. Universelle Data-Wrangling-Werkzeuge vereinen eine grafische Visualisierung, regelbasierte Transformationen und maschinelles Lernen auf einer einzigen Plattform. Daher streben sie an, eine vielfältige Kombination von Vorteilen für eine performante Analyse und Bereinigung zu bieten. Diese Technologien sind in der Regel sowohl als Cloud- als auch als Desktoplösungen verfügbar. Das erlaubt eine Kontrolle über Datenhoheit, Sicherheit und Infrastruktur und bietet zudem eine optimale Skalierbarkeit. Es kann in bestimmten Situationen effizienter sein, bestehende Systeme mit spezialisierten Erweiterungen zu ergänzen, anstatt eine komplette Plattformlösung zu implementieren. Beispielsweise werden Automatisierungskomponenten wie Microsofts „Data Wrangler“ für Visual Studio Code oder TensorFlows „Simple ML for Sheets“ für Google Sheets in die bestehende Infrastruktur integriert [46, 49].

Es kann sinnvoll sein, mehrere Optionen zu vergleichen, wenn ein Datenmanagementteam vor der Entscheidung steht, geeignete Data-Cleansing-Tools einzuführen. Dies kann durch eine praktische Vergleichsanalyse auf Basis eines Probedatensatzes erfolgen. In dem nächsten

Kapitel wird anhand eines solchen Experiments dargestellt, wie sich die Tools in ihren jeweiligen Kategorien tatsächlich unterscheiden.

3 Experimentelle Analyse

Im zweiten Teil der Arbeit werden drei verschiedene Datenbereinigungsansätze – programmierbasiert, workflow-orientiert und KI-gestützt – praxisnah untersucht. Am Anfang wird eine mangelhafte Testdatenbasis erstellt, geeignete Tools werden ausgewählt und ein Bereinigungsplan wird entwickelt. Die Verfahren werden dann technisch realisiert, und die Ergebnisse werden systematisch anhand definierter Bewertungskriterien verglichen. Zum Schluss erfolgt eine Bewertung der Methoden in Bezug auf ihre Effizienz in datenbezogenen Prozessen.

3.1 Methodische Basis: Testdaten, Tools und Vorgehen

Diverse konzeptionelle Fragen zur Auswahl der Testdaten, der zu untersuchenden Instrumente und des allgemeinen Ablaufs des Experiments werden formuliert und beantwortet, bevor es durchgeführt wird. Die Forschungsziele der Arbeit bestimmen die Anforderungen, die richtungsgebend für die Auswahl und Gestaltung des methodischen Rahmens wirken.

3.1.1 Testdatengenerierung

Für die Zielsetzung des Experiments ist die Auswahl geeigneter Rohdaten entscheidend, um Analyseergebnisse zu erhalten, die der Realität möglichst nahekommen. In der Planungsphase werden daher gezielt Merkmale mangelhafter Datenqualität berücksichtigt.

Auswahlstrategie und Motivation des synthetisch generierten Testdatensatzes

Ein entscheidender Aspekt für die zielgenaue Analyse von Datenbereinigungsverfahren ist eine vielfältige Problemstruktur der bearbeiteten Datenbasis. Vor allem wird damit eine genügende Anzahl unterschiedlicher Inkonsistenzen verstanden, die für reale Datensätze charakteristisch sind. Dies ermöglicht eine angemessene Anwendung und Bewertung verschiedener Bereinigungsschritte innerhalb eines einzigen Prozesses. Der Datensatz muss beispielsweise fehlende

Werte, inkonsistente Formatierungen, (unscharfe) Duplikate, Ausreißer sowie fehlerhafte Einträge aufweisen, um regelbasierte, statistische und modellgestützte Ansätze zu evaluieren.

Die Daten müssen zudem sowohl strukturell als auch inhaltlich vielfältig sein: Die Komplexität des Schemas und die Vielfalt der Datentypen (numerisch, kategorial, zeitbezogen) sind zentrale Voraussetzungen für dieses Experiment. Ein Vergleich komplexer Bereinigungsverfahren erfordert unter anderem, dass vielschichtige Abhängigkeiten zwischen Attributen abgebildet oder einzelne Merkmale im Datensatz mehrstufig hergeleitet werden. Darüber hinaus gewährleisten wirtschaftlich relevante Dimensionen und Kennzahlen ein im Geschäftskontext nachvollziehbares Ergebnis sowie eine methodisch saubere Analyse.

Mit dem Ziel, alle benannten Anforderungen in einem einzigen Datensatz zu erfüllen, wird bewusst ein eher kleiner, aber hochvariabler Datensatz synthetisch erzeugt. Dadurch kann das Experiment – im Vergleich zu bestehenden Datenproben aus öffentlichen Quellen (z. B. Kaggle [24]), die häufig nur eine begrenzte Anzahl oder spezifische Anomalien enthalten – effektiver gesteuert werden. Es wird erwartet, dass die Generierung Daten hervorbringt, die vollständig im Rahmen dieser wissenschaftlichen Arbeit liegen und eine aussagekräftige Grundlage für eine praxisnahe Fallstudie bieten.

Aufbau und Eigenschaften des synthetischen Datensatzes

Mit dem Ziel, eine Datenstichprobe zu erstellen, kommen die spezialisierten Python-Bibliotheken „Faker“ und „Random“ zum Einsatz. Die Ergebnisse werden in tabellarischer Form mit der „DataFrame“-Funktion von „Pandas“ weiterbearbeitet [36]. Es wird eine Nutzerdatenbank simuliert, die 1000 Datensätze und 12 funktional verknüpfte Attribute zur Beschreibung der Nutzer eines Online-Dienstes (wie z. B. eines Fitness- oder Lernkurses) enthält.

Die Struktur der synthetisch für die Analyse erzeugten Daten ist in Abbildung 5 durch ein Diagramm dargestellt. Die Entitäten „Customer_Data“ und „Contact_Data“ stehen über den fünfstelligen Integer-Schlüssel „Customer_ID“ in einer 1:n-Beziehung zueinander. „Customer_Data“ umfasst primäre Kundeninformationen. Finanzielle Kennzahlen, die für die Datenanalyse von Bedeutung sind, werden hier zusammen mit persönlichen Angaben (etwa wie Name, Telefonnummer, Geburtsdatum und Alter) erfasst. Dazu zählen der Einkommenswert des Kunden, die Dauer der Kundenbeziehung sowie die Abonnementtypen und deren Werte.

Der zweite Datensatz beinhaltet Korrespondenz-Adressen, die aus unterschiedlichen Gründen – wie dem Schutz sensibler Daten, der Handhabung historischer Anpassungen oder der Komplexität der Erhebung dieser Informationen aus Primärquellen – separat und in einem anderen Format vom ersten Datensatz gespeichert werden können. Es ist vorgesehen, dass mehrere physische oder elektronische Adressen für einen Kunden gespeichert und durch eine eindeutig vierstellige numerische Kennung – „Contact_ID" – historisiert werden können.

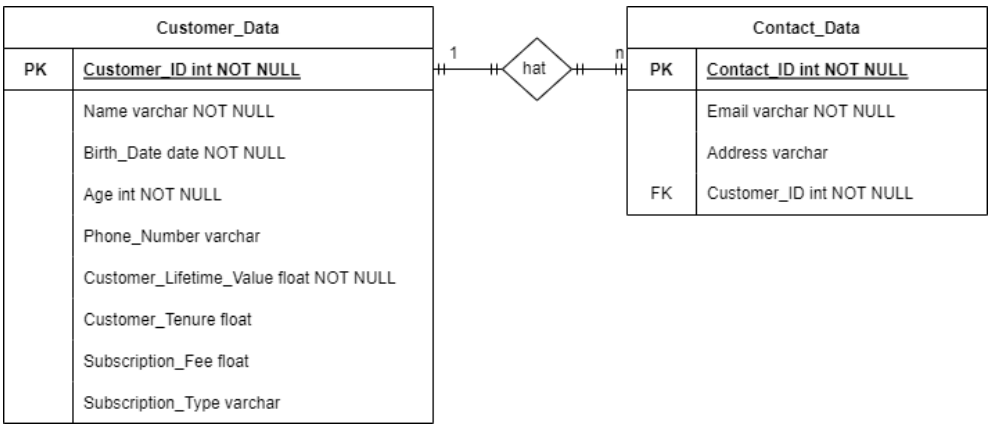


Abbildung 5: ER-Diagramm des Testdatensatzes.

Auf der Schemaebene werden vor allem mögliche Inkonsistenzen wie Duplikate oder fehlende Primärschlüssel dargestellt, die den Prozess der Zusammenführung von Entitäten in ein einziges System erschweren. Attribute, deren Namen inkompatibel sind oder die Überladungen aufweisen, verdeutlichen, dass eine weitere Verarbeitung und Analyse erschwert wird.

Tabelle 2 bietet einen vollständigen Überblick über die synthetisierten Attribute und gibt eine kurze Beschreibung, ein Beispiel aus der Tabelle sowie einen Hinweis auf mögliche „Verunreinigungen" dazu. Trotz der zufälligen Generierung bildet diese Datenstruktur eine nahezu reale logische Beziehung ab. Es existieren sowohl offensichtlichere Zusammenhänge wie das Alter und das Geburtsdatum oder die Höhe und die Art des Abonnements als auch komplexere. Dazu zählen die Beziehungen zwischen der Höhe des Abonnements und der Dauer der Kundenbeziehung sowie die Ermittlung des Kundenwerts anhand dieser Kennzahlen.

Tabelle 2: Übersicht der Testdatenattribute mit gezielten Inkonsistenzen.

Name	***Typ***	***Beschreibung***	***Qualitätsproblemen***	***Beispiel***
Customer ID	Integer	Eindeutiger Identifikator des Kunden	Doppelte oder fehlende Angaben	54950
Name	String	Vollständiger Name	Unterschiedliche Groß-/Kleinschreibung, unscharfe Dubletten oder Datenlücken	Lisa Weaver
Contact ID	Integer	Eindeutiger Identifikator für Korrespondenzkontakte	Doppelte oder fehlende Angaben	1005
Email	String	E-Mail-Adresse	Redundante oder fehlende Einträge pro Kundendatensatz	lisa.weaver77@google.com
Birth Date	Datum	Geburtsdatum	Fehlende oder unplausible Werte	05.05.2000
Phone Number	String	Telefonnummer	Inkonsistente Formatierung oder fehlende Informationen	+1 (212) 198-4989
Age	Integer	Alter, berechnet aus dem Geburtsdatum	Ausreißer oder fehlende Angaben	25
Address	String	Zusammengesetzte Adresse mit Straße, Stadt, Bundesstaat und PLZ	Abkürzungs-, Formatabweichungen oder Datenlücken	46 Daniel Parkways Blvd, New York City, NY 10001
Customer Lifetime Value (CLV)	Float	Berechneter Kundenwert basierend auf Gebühr, Laufzeit und Anpassungsfaktoren	Extremwerte oder inkonsistente Berechnungen	8726.64
Customer Tenure (in years)	Float	Dauer der Kundenbeziehung in Jahren	Ungültige oder fehlende Werte	3.5
Subscription Fee (per month)	Float	Monatliche Lizenzgebühr	Widersprüchliche oder fehlende Angaben	199

Die Logik, die zur Erstellung von Datenfeldern dient, wird weiter ausgeführt.

Der Customer Lifetime Value (CLV) ist eine Schlüsselkennzahl in der Studie zur Erforschung und Vorhersage des Kundenverhaltens. Er wird vereinfacht auf Basis der verfügbaren Attribute auf Jahresbasis (d. h. 12 Monate) berechnet:

$$Customer\ Lifetime\ Value\ =\ Subscription\ Fee\ \times\ 12\ \times\ Customer\ Tenure$$

Die Formel gilt, wenn der Abonnementwert dem durchschnittlichen Umsatz des Kunden entspricht und die Dauer der Geschäftsbeziehung in Halbjahren angegeben wird (z. B. 0,5, 1, 1,5 usw.). Um es realistisch abzubilden, wird in den folgenden Fällen von den Werten abgewichen:

- „Customer Tenure“ beträgt weniger als ein Jahr: Es wird davon ausgegangen, dass der Kunde das Produkt bisher nicht lange genug verwendet hat, um den „Wert“ für das Unternehmen zu stabilisieren. Daher werden das Risiko einer möglichen Kundenabwanderung berücksichtigt, ebenso wie Rabatte und Kosten für gezielte Marketingkampagnen für diesen Zeitraum in Höhe von 15 % des gesamten CLV.
- Premium-Abonnement: Ein hoher Wert des Produkts kann auf Kundenzufriedenheit, Langlebigkeit der Geschäftsbeziehung und eine Steigerung der Gesamtrentabilität hindeuten. Auf diese Weise wird eine Prämie von bis zu 20 % des ursprünglichen CLV erzielt [27].

Diese Messungen werden dann auf der Grundlage der Ergebnisse der Datenbereinigung in den Analysen verglichen.

Die Testdatengenerierung umfasst unter anderem Mustervariationen, zusätzliche Berechnungen und gezielte Manipulationen zur künstlichen Verfälschung. Daher werden die nachstehenden Werteinkonsistenzen modelliert:

- Zwischen 20 % und 35 % der Werte fehlen in verschiedenen Feldern der persönlichen Informationen. Dies spiegelt das Problem der Unvollständigkeit solcher Daten im realen Leben wider.
- Die Formate der Datensätze „Name“, „Phone Number“ und „Address“ sind inkonsistent: verschiedene Groß- und Kleinschreibung von Namen, unterschiedliche Gestaltung von Telefonnummern und Erfassung von physischen Adressen.

- Absolute sowie unscharfe Doppeleinträge: Mehrere Einträge mit demselben Namen enthalten Tippfehler oder Ungenauigkeiten, die zu Fehlinterpretationen führen.
- Ausreißer in „Age“ und „Customer_Tenure“: Abweichungen der Werte von den signifikanten Mittelwerten können das Risiko erhöhen, dass die Analyseergebnisse fehlerhaft sind.
- Die „Customer_Tenure“-Werte werden nach dem Speichern im CSV-Format aufgrund einer automatischen Korrektur fälschlicherweise als Datum interpretiert.

Angesichts der oben beschriebenen zusätzlichen Ungenauigkeiten auf Schemaebene ist das Problem der Verunreinigung solcher Daten erheblich genug, um einen vielschichtigen Bereinigungsprozess zu entwickeln und zu untersuchen. Dieser Ansatz erfüllt nicht nur die oben entwickelten Forschungsanforderungen für Testdaten, sondern bietet auch die Möglichkeit einer anschaulichen Untersuchung von Datenvorverarbeitungstechniken an mehreren häufig vorkommenden Beispielen auf einmal.

3.1.2 Toolauswahl

Der nächste entscheidende Schritt bei der Vorbereitung der Analyse ist die Auswahl der geeigneten Datenverwaltungsinstrumente, die die Idee der Studie am besten widerspiegeln. Da der Vergleich der qualitativen und funktionalen Unterschiede im Zentrum steht, fällt die Wahl auf die drei Ansätze, die sich in ihrer Vorgehensweise am stärksten voneinander unterscheiden:

- Programmierbare Datenpipelines mit Python
- Reinigungskomponenten als Teil des ETL-Prozesses in Pentaho Data Integration
- Interaktive Datenmodifikation auf der Dataiku Data Science Studio Plattform

Die erste Methode beinhaltet, ein individuelles Bereinigungsprogramm zu erstellen. Dieses beginnt mit der Datenintegration und Profilerstellung und reicht über die Transformation und notwendige Änderungen bis hin zum Hochladen im gewünschten Format. Die am besten geeigneten Programmiersprachen für solche Aufgaben sind in der Regel Python oder R. Jede von ihnen bietet eine umfangreiche Auswahl an spezialisierten Funktionen für die Datenverarbeitung, die über Open-Source-Bibliotheken wie pandas, numpy, re, sklearn oder dplyr, tidyr, stringr und viele andere verfügbar sind. Die Wahl fällt auf Python, da die Sprache vielseitig

einsetzbar, leicht verständlich und dank der aktiven Community gut dokumentiert ist [48]. Auch die integrierte Unterstützung für Versionskontrolle spricht für ihren Einsatz.

Der visuelle ETL-Ansatz wird als nächster Forschungsgegenstand betrachtet. Er stellt die nächste Stufe in der Hierarchie der Automatisierung von Reinigungsprozessen dar, da er auf der intuitiven Gestaltung von Arbeitsabläufen über eingebettete Funktionsmodule basiert. Bei der Auswahl eines repräsentativen Werkzeugs werden mehrere populäre End-to-End-Lösungen für die Workflow-Orchestrierung in Betracht gezogen. Trotz der großen Konkurrenz durch grafisch anspruchsvolle Plattformen fällt die Wahl auf Pentaho von Hitachi Vantara: Im Gegensatz zu ähnlichen Anwendungen stellt Pentaho etwa das ETL-Modul getrennt von der BI dar und bietet alle Standardfunktionen einer klassischen Low-Code-Oberfläche [14]. Dies ermöglicht eine optimale Bewertung der untersuchten Datenbereinigungsmethode.

Schließlich werden für die vergleichende Bewertung unter anderem durch maschinelles Lernen realisierte Lösungen angewandt und in der Praxis untersucht. Das Hauptaugenmerk solcher Tools liegt in der Regel auf der Erstellung eines optimierten Modells durch iteratives Lernen und Anpassen. Unter einer Vielzahl solcher DSML-Plattformen (Data Science and Machine Learning) ist Dataiku Data Science Studio hervorzuheben [6]. Ähnlich wie Trifacta Dataprep kombiniert Dataiku interaktive Visualisierung und Anpassung durch die Integration standardmäßiger Data-Science-Module [1]. Der Arbeitsablauf greift auf ein Bündel verschiedener Module zurück, die Standard- und Statistikfunktionen, vorgefertigte AutoML-Plugins oder kundenspezifische Erweiterungen in Form von Codeskripten enthalten. Besonders interessant an Dataiku ist der eingebaute intelligente Assistent auf Basis generativer KI, der die Erstellung betrieblicher Abläufe durch die textuelle Formulierung von Instruktionsprompts erheblich vereinfacht.

Im Rahmen dieses Experiments werden diese drei Instrumente am Beispiel eines einzigen Workflows nicht nur hinsichtlich ihrer Funktionalität, sondern auch in Bezug auf ihre allgemeine Qualität verglichen.

3.1.3 Aufbau des allgemeinen Bereinigungsalgorithmus

Vor der unmittelbaren Entwicklung von Lösungen wird der gesamte Datenverarbeitungsprozess in Form eines einheitlichen Algorithmus modelliert. Ziel dieses Schrittes ist die Erstellung

eines Arbeitsplans, ohne die Entwicklungsumgebung im Vorfeld zu berücksichtigen. Angesichts der Unterschiede der analysierten Technologien wird gleichzeitig beachtet, dass es bei der direkten Implementierung zu Abweichungen von diesem Plan kommen kann. Beispielsweise kann eine Änderung der Funktionsausstattung oder die Kombination mehrerer Aktionen in einer Phase nicht nur die Reihenfolge bestimmter Schritte, sondern auch die Dauer und Komplexität des gesamten Prozesses beeinflussen.

Unter Berücksichtigung der jeweiligen Vorteile und Einschränkungen verschiedener Bereinigungsmethoden wird der logische Rahmen des Prozesses als optimale Reihenfolge von Schritten im Anhang B modelliert. Der hier beschriebene Algorithmus stellt einige der zentralen Maßnahmen zur Behebung von Problemen auf Datensatz- und Schemaniveau dar [22].

Sobald die Daten eingehen, wird zunächst festgelegt, wie sie in das Arbeitssystem integriert werden sollen. Bei mehreren Quellen wird die Kompatibilität geprüft und gegebenenfalls werden Schemata für die logische Zusammenführung der Daten standardisiert. Wenn die Daten im System erfasst sind, erfolgt eine Untersuchung auf Ungenauigkeiten. Dies schließt sowohl die technische Analyse als auch die Überprüfung der fachlichen Anforderungen ein.

Nach der Analyse und Identifizierung von Problembereichen im Datensatz geht es zunächst darum, Formate und Feldtypen für eine optimale Weiterverarbeitung zu standardisieren. Dabei geht es darum, widersprüchliche Datensätze in eine sinnvoll kohärente Form zu bringen, wobei die Besonderheiten von textuellen, numerischen und kategorialen Informationstypen zu berücksichtigen sind.

Daraufhin wird eine Wertebereinigung in mehreren Stufen durchgeführt: Nicht aussagekräftige oder leere Zeichenfolgen werden entfernt und überladene Attribute werden umgewandelt. Ferner werden Duplikate nach dem Prinzip der Fuzzy-Logik und Validierungsmechanismen auf allen Ebenen entfernt: sowohl zeilenweise als auch innerhalb bestimmter Cluster. Eine weitere Aufgabe ist die Identifizierung und der Ausschluss von Extremwerten aus Zahlenreihen, beispielsweise mithilfe statistischer Methoden.

Die letzte Aktivität im Prozess ist das Auffüllen der fehlenden Werte im normalisierten und standardisierten Datenblock. Werte, die fehlen und deren Einfluss auf die Analyseergebnisse nicht erheblich ist, werden durch Platzhalter ersetzt. Haben sie jedoch einen Einfluss, so werden

sie je nach Komplexität der funktionalen Abhängigkeiten mit geeigneten Methoden berechnet. In Fällen, in denen es eine direkte und eindeutige Abhängigkeit gibt, kann man Gruppierungen oder Clustering mit Mittelwertsuche sowie andere Berechnungen vornehmen. Wenn mehrere Faktoren gleichzeitig Einfluss nehmen, sind komplexe Modelle basierend auf Regression, Gradient Boosting und anderen trainierten Modellen möglich.

Am Ende des Zyklus wird das Bereinigungsergebnis erneut validiert. Je nachdem, ob die Daten als fehlerfrei erkannt wurden oder nicht, werden sie entweder im erforderlichen Format hochgeladen oder weiterverarbeitet. In einigen Fällen kann es sinnvoll sein, in den folgenden Iterationen neue Reinigungsmechanismen oder manuelle Einstellungen vorzunehmen.

3.2 Technische Umsetzung der Bereinigungsverfahren

Im nächsten Kapitel wird die praktische Entwicklung eines allgemeinen Bereinigungsprozesses beschrieben. Dazu werden für jeden Ansatz entsprechende Tools verwendet. So wird der Prozess mithilfe von Python-Pipelines, einem sequenziellen ETL-Workflow sowie einer intelligenten Datenmanagement-Plattform umgesetzt.

3.2.1 Entwurf mit Python

In Python wird das Cleansing-Framework auf eine funktionsbasierte Weise implementiert. Jupyter Notebook, eine Open-Source-Webanwendung, wird verwendet, um den Code detailliert zu untersuchen und schrittweise zu bearbeiten. In dieser Umgebung können Pipelines einzeln ausgeführt und Zwischenergebnisse betrachtet werden. Selbst ohne spezifische grafische Oberfläche ist es möglich, den gesamten Entwicklungsprozess nachzuvollziehen. Mit diesem Entwurf soll ein automatisiertes Programm entwickelt werden, das Rohdaten aus internen Ressourcen bereinigt und konsolidiert.

Der Workflow wird durch die Erzeugung der Klasseninstanz „CustomerDataCleaner“ gestartet, wobei die Dateipfade für Input- und Outputdaten übergeben werden (Listing 1).

Listing 1: Struktur der Ausführungspipeline des Frameworks.

```
class CustomerDataCleaner:

	def __init__(self, base_path, addresses_path, output_path):
		self.base_path = base_path
		self.addresses_path = addresses_path
		self.output_path = output_path
		self.merged_df = None
		self.uniformed_df = None
		self.deduplicated_df = None
		self.valid_df = None
		self.completed_df = None

	def run(self):
		start_time = time.time()
		try:
			self.load()
			self.merge()
			self.standardize()
			self.deduplicate()
			self.clean_outliers()
			self.impute_missing_values()
			self.finalize()
		except Exception as e:
			print(f"Pipeline fehlgeschlagen: {e}")
		finally:
			end_time = time.time()
			execution_time = end_time - start_time
			print(f"Gesamtzeit der Pipeline-Ausführung: {
				execution_time:.2f} Sekunden"
			)
```

Das Grundprinzip besagt, dass die aktuellen Zustände der Datensätze auf Instanzenebene (wie „merged_df“, „uniformed_df“ usw.) kontinuierlich gespeichert werden. Zuerst wird vor den nächsten Anpassungen der Daten überprüft, ob diese ihre Integrität bewahren: Es wird geprüft, ob der im vorherigen Schritt benötigte DataFrame bereits existiert. Dadurch wird sichergestellt,

dass die geplante Pipeline-Kette korrekt ist und die einzelnen Bereinigungsschritte unabhängig voneinander durchgeführt und bewertet werden können.

Jeder Bereinigungsschritt wird durch eine eigene Methode repräsentiert.

Als Erstes werden die Rohdaten aus den übergebenen Datenpfaden geladen und in einem kompakten DataFrame abgelegt. Automatisch erfolgt dies, indem die Pandas-Funktionen read_csv() und read_json() für die Kunden- bzw. Kontaktdatensätze aufgerufen werden. Die zwei Komponenten werden zu einem einzigen Objekt zur Verarbeitung zusammengeführt. Zuerst wird mit der speziellen Funktion analyze() eine Überprüfung gestartet. Die Daten werden auf die Konsistenz ihrer Schlüssel gemäß den Integrationsrichtlinien überprüft (Listing 2).

Listing 2: Funktion zur Erstellung von Datenqualitätsprotokollen.

```
def analyze (self, df, name):
       print(f"--- Analysebericht für {name} ---")
       print("\nGrundlegende Informationen:")
       df.info()
       print("\nStatistische Kennzahlen:")
       display(df.describe(include="all"))
       print("\nFehlende Werte pro Spalte:")
       display(df.isnull().sum().to_frame(name='Anzahl fehlender Wer-
te'))
       print(f"\nAnzahl der duplizierten Zeilen: {df.duplica-
ted().sum()}")
       if df.duplicated().sum() > 0:
              print("\nDuplizierte Zeilen:")
              display(df[df.duplicated()].head())
       else:
              print("\nKeine exakten Duplikate gefunden.")
```

Dank der Operationen describe() und info() wird eine detaillierte Beschreibung der Menge, des Formats und der primären statistischen Kennzahlen jedes Feldes der Datenobjekte erzeugt. Daraufhin werden die „Customer_ID“-Schlüssel in den Datentyp Int64 umgewandelt, ein Left-Merge unter Berücksichtigung des Designs durchgeführt und neue Schemata standardisiert.

Daraufhin wird der neue Datenrahmen analysiert. Auf Grundlage der Resultate wird eine primäre sequenzielle Standardisierung sämtlicher Felder vorgenommen. Datensätze mit Lücken in den ID-Schlüsseln, dem Namen oder der E-Mail, die nicht wiederhergestellt werden können, werden gelöscht. Die anschließende Verarbeitung erfolgt optimal mit den für die Analyse zentralen Informationen: Datenfelder werden manipuliert, um sie spaltenkonsistent in ein geeignetes Format zu bringen. Beispielsweise werden Alter, Kundenbeziehungsdauer und Kundenwert entsprechend der Verwendungslogik als Int64 und Float64 gespeichert. Mit str.replace() werden Einträge, die keine Zahlen sind, in Fließkommazahlen umgewandelt (z. B. „3. Mai“ wird zu 3.5). Außerdem werden die textuellen Daten mit den String- und Regex-Funktionen korrigiert und standardisiert. Namen werden mit str.title() und str.split() als Vor- und Nachnamen in Titelschreibung gespeichert und Telefonnummern mit re.sub() und str.lstrip() auf den Typ „+1 XXX-XXX-XXXX“ für US-Nummern eingestellt (Listing 3).

Listing 3: Pipeline zur Standardisierung von Telefonnummern.

```
def standardize_phone(phone):
       if pd.isna(phone) or str(phone) == "":
              return phone
       phone = re.sub(r"\D", "", str(phone))
       phone = phone.lstrip("1") if len(phone) == 11 else phone
       return f"+1 {phone[:3]}-{phone[3:6]}-{phone[6:]}"
self.uniformed_df["Phone_Number"] = self.uniformed_df["Phone_Number"
].astype("string").apply(standardize_phone)
```

In Adressen werden die Straßenabkürzungen durch eine Pattern-Suche verkürzt, und die Einzelteile werden zuletzt als „Street“, „City“, „State“ und „Zip_Code“ extrahiert. Die Datenstandardisierung für das Geburtsdatum erfolgt mittels dt.strftime().

Im vierten Schritt werden die verarbeiteten Daten dedupliziert. Zu Beginn werden exakte Zeilenduplikate durch die automatisierte Pandas-Funktion drop_duplicates() entfernt. „Unscharfe“ Duplikate im Namensfeld werden anschließend ermittelt und durch das Geburtsdatum validiert. Es erfolgt eine iterative Analyse einzelner Zeilen mit der Funktion ratio() aus der Bibliothek „rapidfuzz“ [37]: Der Ähnlichkeitswert zwischen zwei Namen wird mit einem Threshold von

75 % verglichen. Das Resultat ist ein Duplikatensatz, der aus Paaren von Customer_ID und Name zusammengesetzt ist. Es wird vorgesehen, aus jedem Paar nur die Customer_ID mit dem höchsten Contact_ID-Wert zu bewahren, um sicherzustellen, dass die aktuellsten Kontaktinformationen des Kunden vorhanden sind.

Nach der Standardisierung und Deduplication der Daten werden Ausreißer korrigiert. In diesem Schritt werden die Werte in den Spalten „Age“, „Customer_Tenure“ und „Customer_Lifetime_Value“ auf auffällige Extremwerte untersucht. Dies beinhaltet zum Beispiel, dass die Alterseingaben negativ oder geringer als der Kundenwert im Datensatz auffallen. Eine der Methoden ist das IQR-Verfahren (Interquartilsabstand), welches die Wertgrenzen mit den folgenden Formeln berechnet [9, S. 162, 190]:

$$Untergrenze = 25.Perzentil - 1.5 * Interquartilsabstand$$

$$Obergrenze = 75.Perzentil + 1.5 * Interquartilsabstand$$

Diese Parameter werden innerhalb der Methode find_iqr_outliers() mithilfe von quantile() und anderen mathematischen Funktionen der Bibliothek „Pandas“ kalkuliert (Listing 4). Alle Werte, die außerhalb der berechneten Schranken liegen, werden für jede analysierte Spalte erkannt und ihre Indizes werden im Set „iqr_outliers“ abgelegt. Anschließend werden alle ermittelten Ausreißer mittels Pandas-Funktion drop() gesammelt aus dem Datensatz entfernt.

Listing 4: Ausreißerbereinigung anhand des IQR-Verfahrens.

```
def find_iqr_outliers(df, cols):
       iqr_outliers = set()
       for col in cols:
              Q1, Q3 = df[col].quantile([.25, .75])
              IQR = Q3 - Q1
              lower_bound = Q1 - 1.5 * IQR
              upper_bound = Q3 + 1.5 * IQR
              iqr_outliers.update(df.index[(df[col] < lower_bound) |
                     (df[col] > upper_bound)])
       return iqr_outliers
```

Um den Erfolg dieses Schritts nachzuvollziehen, wird die Interquartilsverteilung dieser Spalten vor und nach der Bereinigung mithilfe der Bibliothek „Matplotlib“ in Form von Boxplots

visualisiert. Da die Werte weit gestreut sind, wird eine symmetrisch logarithmische Skalierung verwendet (Abbildungen 6 und 7). Es ist somit erkennbar, dass negative oder extrem hohe Werte in den Spalten „Age" und „Customer_Tenure" erfolgreich entfernt werden. Zudem wird für die Spalte „Customer_Lifetime_Value" der Unterschied zwischen dem Minimalwert und dem 25. Quantil durch die Normalisierung anderer Attribute deutlicher hervorgehoben.

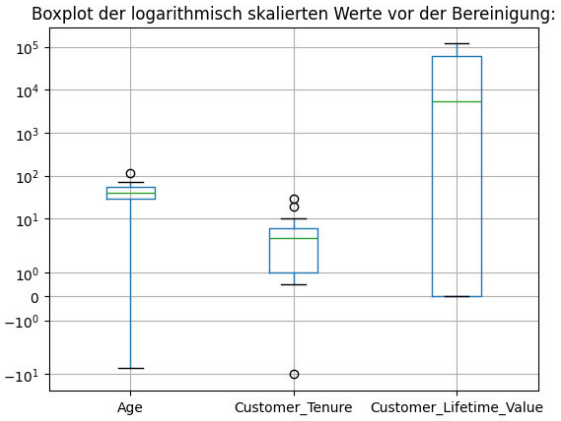


Abbildung 6: Boxplot der numerischen Felder vor der Ausreißerbereinigung.

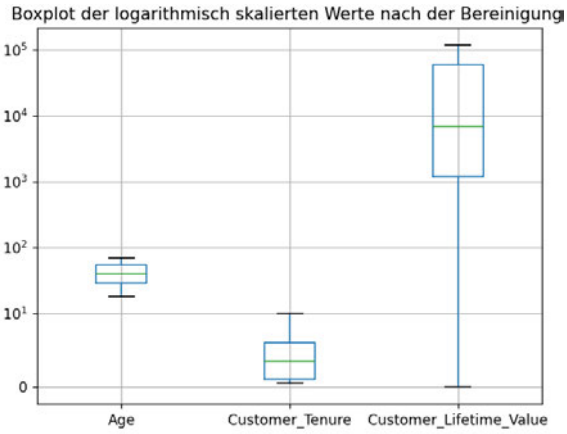


Abbildung 7: Boxplot der numerischen Felder nach der Ausreißerbereinigung.

Die letzte Bereinigungsmaßnahme in dem Prozess ist die Behandlung fehlender Werte. Die Strategie hierfür unterscheidet sich je nach Feld und dessen logischer Abhängigkeit. Bei einem vorhandenen Geburtsdatum lässt sich beispielsweise das Alter berechnen. Wenn für Spalten jedoch keine alternativen Werte vorhanden sind (z. B. „Address" oder „Phone_Number"), werden deren Lücken mit sinnvollen Platzhaltern wie „Not provided" gefüllt und bei der Analyse berücksichtigt. Da die Abonnementgebühren direkt zu einem konkreten Abonnementtyp

gehören, können fehlende Werte hier ganz genau durch einen Modus einer zugewiesenen Gruppe definiert werden. Ein passender Wert für die multiabhängige Messung, wie „Customer_Tenure“, wird zwar komplizierter, aber präziser gefunden. Dafür wird das überwachte Machine-Learning-Modell k-nearest neighbors (kNN) mittels KNNImputer() aus der Bibliothek sklearn.impute angewendet [43]. Wie es aus dem Forschungsdesign folgt, hängt die Kundenbeziehungsdauer vom Kundenwert und vom Abonnementwert ab. Also werden all diese Informationen für die KNN-Imputation angewendet. Die Anzahl der zu vergleichenden Nachbarwerte wird auf zwei festgelegt, um die lokale Vorhersage durchzuführen (Listing 5). Die Ergebniswerte werden abschließend modifiziert und im passenden Format als Fließkommazahlen gespeichert.

Listing 5: Auffüllen der fehlenden Werte in „Customer_Tenure“ mittels KNN-Methode.

```
impute_data = df[["Customer_Tenure", "Customer_Lifetime_Value", "Sub-
scription_Fee"]]
imputer = KNNImputer(n_neighbors=2)
imputed_values = imputer.fit_transform(impute_data)
df["Customer_Tenure"] = (imputed_values[:, 0]*2).round()/2
```

Der Prozess wird durch die Korrektur der Feldreihenfolge, die abschließende Analyse und das Hochladen der Ergebnisse im CSV-Format abgeschlossen.

3.2.2 Entwurf mit Pentaho DI

Im Gegensatz zu vollständig manuell programmierbaren Bereinigungssystemen basiert die zweite eingesetzte Technologie auf einer Abfolge grafischer Funktionseinheiten. Diese einzelnen Schritte sind innerhalb einer sogenannten „Transformation“ sequenziell oder parallel miteinander verknüpft. Die Kette solcher transformierender Teilprozesse bildet – zusammen mit definierten Start- und Endkomponenten – eine umfassende, lineare Low-Code-Lösung in Form eines „Jobs“. Der in Pentaho implementierte Datenfluss ist in Abbildung 8 dargestellt.

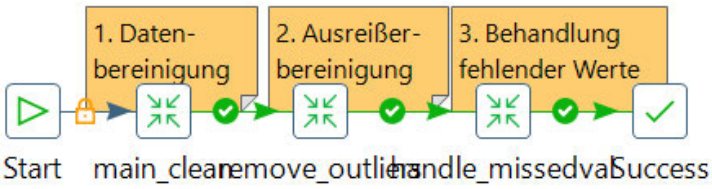


Abbildung 8: Ablauf des Datenbereinigungs-Jobs in Pentaho.

Drei grundlegende Transformationen bilden den Workflow: die Hauptbereinigung, das Finden und Entfernen von Ausreißern sowie das Schließen von Datenlücken. Die Aufteilung ist eine Folge der Zwischenspeicherung und -verarbeitung der Daten in einer separaten Umgebung, um die einzelnen Schritte effizienter gestalten zu können.

Die meisten Verarbeitungsschritte sind in der Transformation „main_clean“ zusammengefasst (Abbildung 9). Der Import von Daten erfolgt über Vorlagen für CSV- und JSON-Dateiformate, die Metadaten wie Typ, Format oder Wertlänge festlegen. Der Datenbestand wird nach einem linken Join per „Stream lookup“ durch die Komponenten „Unique rows“ und „Data validator“ von Zeilenduplikaten und ungültigen Werten bereinigt. Mit einem einzigen Schritt werden somit fehlende ID-Attribute, E-Mail-Adressen mit falschem Format, inkorrekte Kundenwerte, ungültige Alterseinträge und andere Inkonsistenzen herausgefiltert.

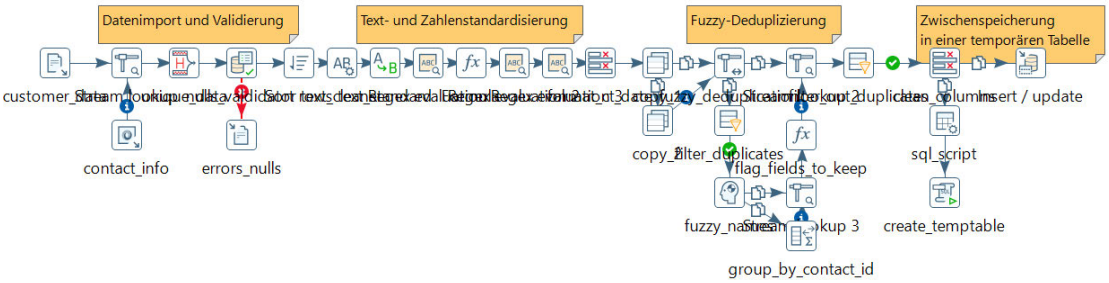


Abbildung 9: Erste Transformationsstufe: Standardisierung und Deduplizierung.

Text und einzelne numerische Daten werden in einer sechsphasigen Sequenz von String-basierten Funktionen und regulären Ausdrücken standardisiert oder in ein gültiges Format überführt. Das Feld „Name“ wird automatisch in das Titelformat umgewandelt, mit den String-Operationen „Lower“ und „InitCap“. In den Komponenten „Regex evaluation“ und „Formula“ werden Telefonnummern, Kundenbeziehungsdauer und Straßenabkürzungen durch mehrere regex-gesteuerte Anpassungen korrigiert.

Im Rahmen dieser Haupttransformation werden unscharfe Duplikate im Feld „Name“ als letzter Schritt durch die integrierte „Fuzzy match“-Funktion von Pentaho bereinigt. Um alle Tupel miteinander zu vergleichen, wird der Datenfluss hierfür dupliziert. Im Vergleich zu „RapidFuzz“ von Python lässt sich in Pentaho die verwendete Ähnlichkeitsprüfung aus mehreren Optionen auswählen. Beim Levenshtein-Verfahren werden die Schwellenwerte als absolute, unabhängig von der Wörterlänge definierte Minimal- und Maximalwerte festgelegt. Nachdem die Namensduplikate anhand des Geburtsdatums validiert wurden, erfolgt die Identifizierung der Einträge, die gelöscht werden sollen: Die Anweisung „Memory group by“ speichert den entsprechenden Eintrag in jeder Duplikatengruppe, indem sie die höchste „Contact_ID“ ermittelt. Die Schritte „Stream lookup“ und „Formula“ dienen dazu, die vorhandenen Duplikate innerhalb der Gruppe zu erkennen und sie aus der Datenbasis zu entfernen.

Um die Performance zu erhöhen, werden die aktuellen Daten in einer temporären Tabelle innerhalb einer eingebetteten H2-Datenbank zwischengespeichert, bevor sie weiterverarbeitet werden. Die nächste Transformation, bei der Ausreißer in mehreren numerischen Attributen durch eine Kombination aus SQL-Anweisungen und benutzerdefinierten Filtern identifiziert und entfernt werden, wird entkoppelt durchgeführt (Abbildung 10).

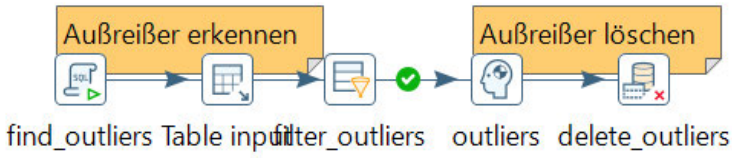


Abbildung 10: Zweite Transformationsstufe: Ausreißerbereinigung.

Pentaho stellt keine funktionalen Module zur Ausreißererkennung bereit, deswegen wird ein geeignetes Verfahren basierend auf dem Interquartilsabstand mithilfe von SQL entwickelt. Die Funktion PERCENTILE_CONT() wird verwendet, um die 25. und 75. Perzentile zu berechnen, welche als untere und obere Grenze für die Extremwertidentifikation dienen (Listing 6). Diese werden im Datensatz gekennzeichnet und über den Schritt „Delete“ entfernt.

Listing 6: Ausreißerbereinigung per SQL-Anweisung im Pentaho-Workflow.

```
ALTER TABLE CUSTOMER_DATA ADD COLUMN IF NOT EXISTS Is_Outlier BOOLEAN
DEFAULT FALSE;
```

```
WITH quartiles AS (
	SELECT
		PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY Customer_Life-
time_Value) AS Q3_CLV
		PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY Age) AS Q1_Age,
		PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY Age) AS Q3_Age,
		PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY Customer_Tenure)
AS Q1_Tenure,
		PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY Customer_Tenure)
AS Q3_Tenure,
		PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY Customer_Life-
time_Value) AS Q1_CLV,
	FROM CUSTOMER_DATA
),
iqr_calculated AS (
	SELECT
		Q1_Age,
		Q3_Age,
		Q3_Age - Q1_Age AS IQR_Age,
		Q1_Tenure,
		Q3_Tenure,
		Q3_Tenure - Q1_Tenure AS IQR_Tenure,
		Q1_CLV,
		Q3_CLV,
		Q3_CLV - Q1_CLV AS IQR_CLV
	FROM quartiles
)
UPDATE CUSTOMER_DATA
SET Is_Outlier = TRUE
WHERE
	Customer_Tenure < 0
	OR Age <= Customer_Tenure
	OR Age < (SELECT Q1_Age - 1.5 * IQR_Age FROM iqr_calculated)
	OR Age > (SELECT Q3_Age + 1.5 * IQR_Age FROM iqr_calculated)
	OR Customer_Tenure < (SELECT Q1_Tenure - 1.5 * IQR_Tenure FROM
iqr_calculated)
```

```
       OR Customer_Tenure > (SELECT Q3_Tenure + 1.5 * IQR_Tenure FROM
iqr_calculated)
       OR Customer_Lifetime_Value < (SELECT Q1_CLV - 1.5 * IQR_CLV FROM
iqr_calculated)
       OR Customer_Lifetime_Value > (SELECT Q3_CLV + 1.5 * IQR_CLV FROM
iqr_calculated);
```

In der letzten Phase des Bereinigungsworkflows werden alle fehlenden Werte in den Spalten entsprechend ihrem Datentyp und der inhaltlichen Logik behandelt. Insgesamt umfasst dieser Unterprozess vier Abschnitte, in denen nacheinander durch den Einsatz von Formeln, Gruppierungen, Filterungen und Lookup-Tabellen realistische Ersatzwerte für die vorhandenen Lücken berechnet und eingesetzt werden (Abbildung 11).

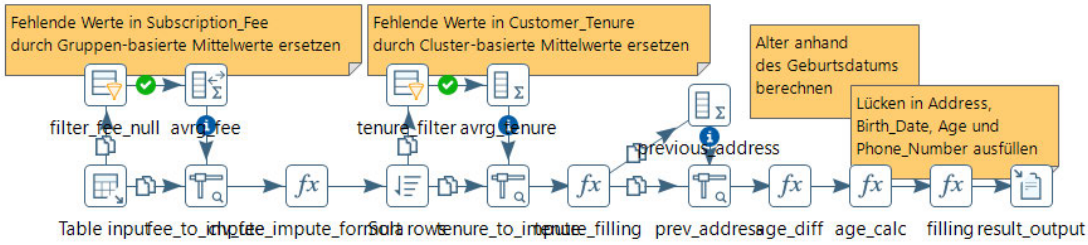


Abbildung 11: Dritte Transformationsstufe: Behandlung fehlender Werte.

Bei der Imputation auf Gruppen- oder Clusterbasis wird stets mit doppelten Datenflüssen gearbeitet. Einerseits wird aus dem aktuellen Datensatz die betroffene Spalte ohne Nullwerte herausgefiltert, um darauf aufbauend – direkt oder unter Nebenbedingungen – einen gruppenbezogenen Durchschnittswert zu berechnen. Weil Pentaho keine Module für fortgeschrittene Imputationsverfahren wie KNN oder andere ML-Modelle hat, wird für „Customer_Tenure“ ein Mittelwert ermittelt, basierend auf einer Clustereinteilung nach Kundenwertgröße und Abonnementgebühr. So wird unter den gegebenen technischen Bedingungen eine möglichst präzise Schätzung ermöglicht. Die Ergebnisse werden in neuen Spalten gespeichert und anschließend mithilfe von bedingten Formeln als Ersatz eingesetzt.

Im Anschluss werden Ersatzwerte für „Address“ und „Age“ bestimmt. Die letzte bekannte Adresse eines Kunden wird durch eine Aggregation nach Kundennummer in der Spalte „Address“ bestimmt, indem man das Prinzip des „Last non-null value“ anwendet. Das Alter berechnet sich aus dem Unterschied zwischen dem Geburtsdatum und dem heutigen Datum.

Endgültig bereinigte Daten werden zurückgegeben, sobald die mit „Replace value“ und „Formel“ gekennzeichneten Lücken durch passende Werte oder Platzhalter ergänzt werden.

3.2.3 Entwurf mit Dataiku DSS

In einer technischen Umgebung vereint Dataiku DSS KI-gestützte Analysen, Datenvisualisierungen und die klassischen ETL-Prozesse. Anders als bei streng sequenziellen Prozessen erlaubt die interaktive Verarbeitung innerhalb eines einzigen Rezepts – einer geordneten Abfolge von Datentransformationen. Erweiterte Funktionen wie das Erstellen und Trainieren eigener ML-Modelle, der Einsatz von Prompt-Anweisungen zur gezielten Steuerung des KI-Systems, die partielle Ausführung von Workflows und die Konsistenzprüfung des Schemas sind dabei möglich. Die entworfene Ablaufstruktur ist dadurch kompakter als in Pentaho und besteht aus rund 15 Verarbeitungsschritten (Anhang C).

Das Programm erkennt während der Datenintegration automatisch alle Metainformationen, wie Datentypen und die Bedeutung der Felder, und speichert sie intern als Schemata. Diese Datenstrukturen werden in allen folgenden Verarbeitungsschritten durchlaufen und zeigen dabei die Änderungen, die an ihnen vorgenommen wurden.

Die Zusammenführung von Datensätzen ist flexibel einstellbar: Duplikate und fehlende Primärschlüssel aus beiden Dateninstanzen können durch eine Vorfilterung herausgefiltert werden. In den Bereichen „Join“ und „Selected Columns“ werden dafür die Verknüpfungskriterien und die Spalten, die ausgegeben werden sollen, gezielt über eine Auswahlliste festgelegt.

Nach jedem Rezept erzeugt Dataiku eine farblich markierte Datenvisualisierung, die auf der Bedeutungsvalidität von Feldern basieren kann. Sie erleichtert zusammen mit einer interaktiven Schnittstelle die Erkennung und Analyse von Datenqualitätsproblemen. Das System stellt wichtige Dateikenngrößen sowie eine detaillierte Übersicht über Spaltenstatistiken bereit und ermöglicht es, für einzelne Attribute Qualitätsregeln festzulegen (Abbildung 12).

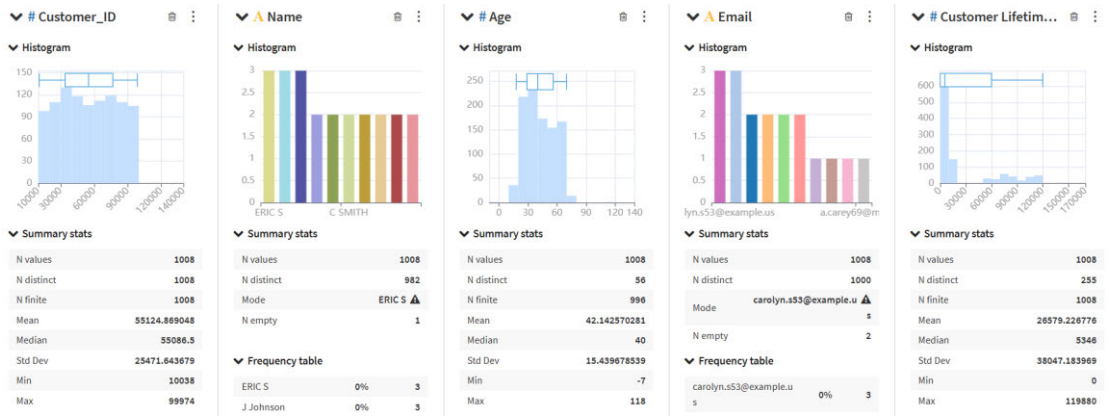


Abbildung 12: Beispiel für eine statistische Datenübersicht in Dataiku.

Basierend auf automatisch erkannten Inkonsistenzen werden zunächst Filter- und Bereinigungsvorgänge angewendet. Im Unterschied zu früheren Lösungen können dabei mehrere Schritte unkompliziert und manuell – etwa durch Drag-and-Drop, integrierte Funktionen wie „Remove rows outside of 1.5 IQR“ oder durch KI-gestützte Generierung mittels textueller Prompts – erstellt werden (Abbildungen 13, 14). Die vom Programm erzeugten Algorithmen sind nachvollziehbar, und ihre Wirkung wird entweder durch eine Vorschau der modifizierten Zeilen oder durch eine statistische Schnellansicht der bereinigten Spalten überprüft.

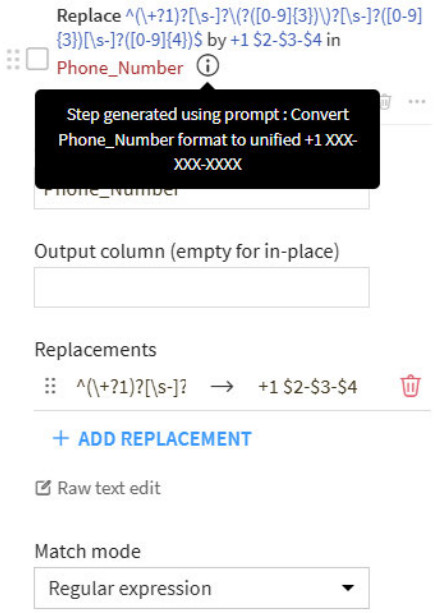


Abbildung 13: Formatierung von Telefonnummern mittels KI-Agenten in Dataiku.

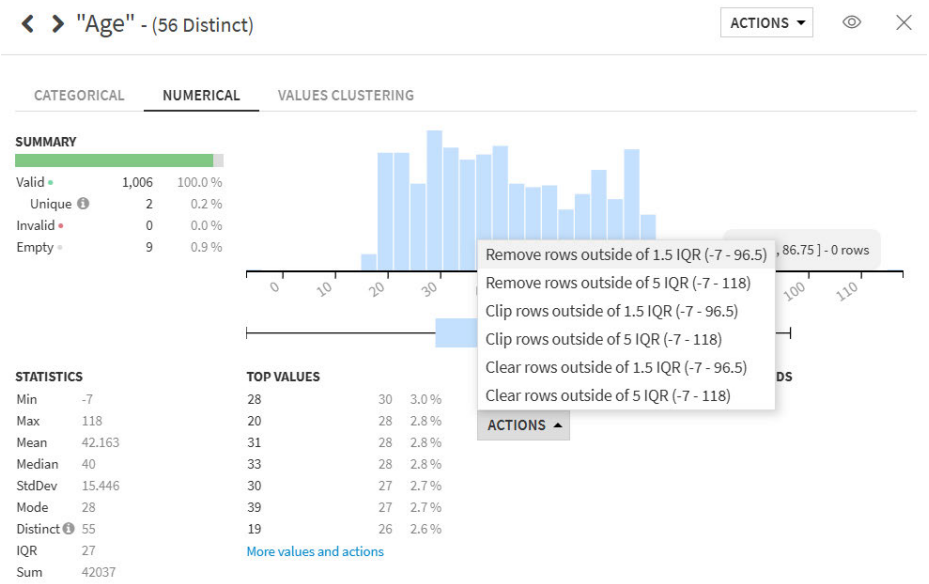


Abbildung 14: Von Dataiku bereitgestellte Ausreißerbereinigung.

Der folgende Schritt im Dataiku-Workflow beschäftigt sich mit dem Identifizieren und Beseitigen unscharfer Duplikate. Dieser Schritt fußt ebenfalls auf zwei separaten Datenströmen. Die Distanz zwischen den Namen wird, wie es auch bei den bereits existierenden Implementierungen in Python oder Pentaho der Fall ist, durch das Damerau–Levenshtein-Verfahren bestimmt. Dabei kommt ein festgelegter Schwellenwert von 75 % zum Einsatz. Bei einem Left-Join werden die „Contact_ID“ des Originals sowie die Kombinationen ähnlicher Werte für die Attribute „Name“ und „Birth_Date“ aus den verglichenen Datenkopien extrahiert. Die Schritte des Gruppierens und Bereinigen kennzeichnen eindeutig die Kontaktidentifikatoren, die den zu entfernenden Duplikateinträgen zugeordnet sind. Die Vorfilterung bei der Aktion „Left anti join“ ermöglicht es, alle Datensätze zu entfernen, deren Verknüpfungswerte mit „false“ gekennzeichnet sind.

Abhängig von der Komplexität der Struktur und der Logik der jeweiligen Attribute werden unterschiedliche Plattformoperationen genutzt, um die Lücken in den Feldern zu schließen. In einfacheren Fällen können die erforderlichen Verarbeitungsschritte innerhalb eines Rezepts entweder komplett durch Künstliche Intelligenz basierend auf einer Textanweisung, wie bei „Subscription_Fee“, „Age“ oder „Birth_Date“, oder durch die Nutzung der integrierten Funktion „Fill empty rows“, wie bei „Phone_Number“, automatisiert erstellt werden. Attribute wie „Address“ und „Customer_Tenure“ haben eine komplexere Logik, weshalb sie mit einer

umfangreichen Konfiguration, die aus mehreren Rezepten besteht, bereinigt werden. Die fehlenden Werte bei den Adressen werden nach dem Prinzip der Historisierung ersetzt: Die Einträge werden anhand der Korrespondenznummer der jeweiligen Kunden zunächst sortiert und gruppiert („First non-null value“), um den ersten nicht-leeren Wert zu finden und danach zu verwenden.

Um die fehlenden Werte in der Spalte „Customer_Tenure“ mittels Regression zu prognostizieren, wird es analog zur Lösung in Python ein Verfahren des maschinellen Lernens verwendet. Sobald vollständige Beispieldaten ausgewählt sind, erstellt, testet und evaluiert die erweiterte Funktion „AutoML Prediction“ mehrere Regressionsmodelle mit vordefinierten Parametern, um Attributbeziehungen abzubilden. Die automatisierte Bewertung der erzeugten Systeme erfolgt anhand von acht grundlegenden statistischen Kennzahlen, darunter der Determinationskoeffizient R^2 (auch als Bestimmtheitsmaß bezeichnet). „Extra Trees“ wird aufgrund dieser Analyseergebnisse als die insgesamt leistungsfähigste und effizienteste Methode zur Imputation von Datensatzlücken identifiziert, während es im Ranking sehr nah am in Python implementierten K-Nearest-Neighbours-Modell ist (Abbildung 15).

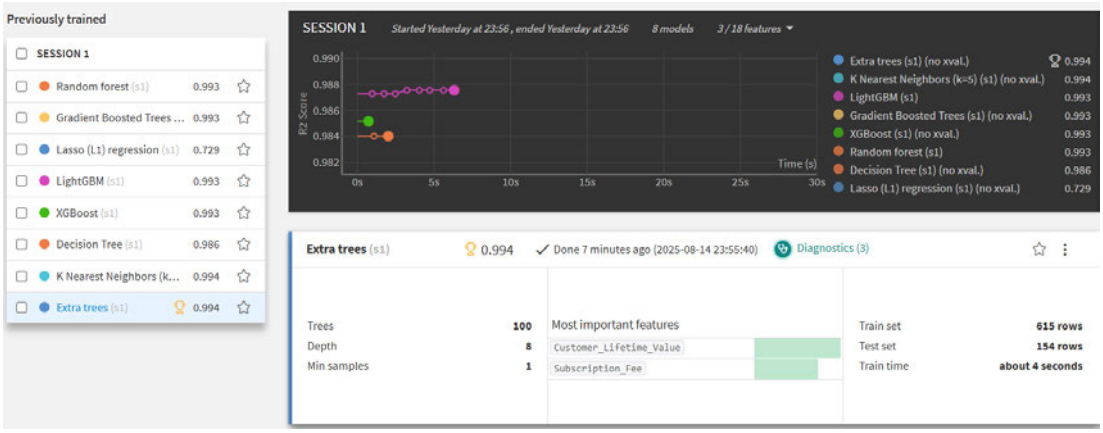


Abbildung 15: ML-Modelle zum Umgang mit fehlenden Werten in Dataiku im Vergleich.

Am Ende des Prozesses werden die prognostizierten „Customer_Tenure“-Werte in das geeignete Float-Format für eine Halbjahresbezeichnung umgewandelt. Das Schema wird danach angepasst, und der Datensatz wird exportiert.

3.3 Evaluierung und Vergleich der Ergebnisse

Die Datenbereinigungsansätze werden in der abschließenden Phase des Experiments anhand der hier verwendeten Werkzeuge und der implementierten Beispielsalgorithmen analysiert. Die Resultate werden durch eine Bewertungsmatrix, eine Evaluierung der CPU-Auslastung und Laufzeit sowie einen qualitativen Vergleich von einzelnen Teilprozessen präsentiert. Die Messungen erfolgen auf einem Rechner mit einem 11th Gen Intel® Core™ i7-1195G7 2,90-GHz-Prozessor.

Zuerst erfolgt die Bewertung der Untersuchungsobjekte anhand verschiedener Kriterien und der Benutzererfahrungen, die im Laufe des Experiments gesammelt wurden. Hierzu gehören unter anderem die Breite der Funktionalitäten, Usability, Integrität, Anpassungsfähigkeit an persönliche Bedürfnisse, Komplexität und weitere Faktoren, die durch die bestehende Literatur festgelegt werden [8, 30, 34]. Abbildung 16 zeigt die Ausprägungen von zehn Hauptfaktoren in Form einer Matrix auf einer Skala von 1,0 (niedrig) bis 5,0 (hoch).

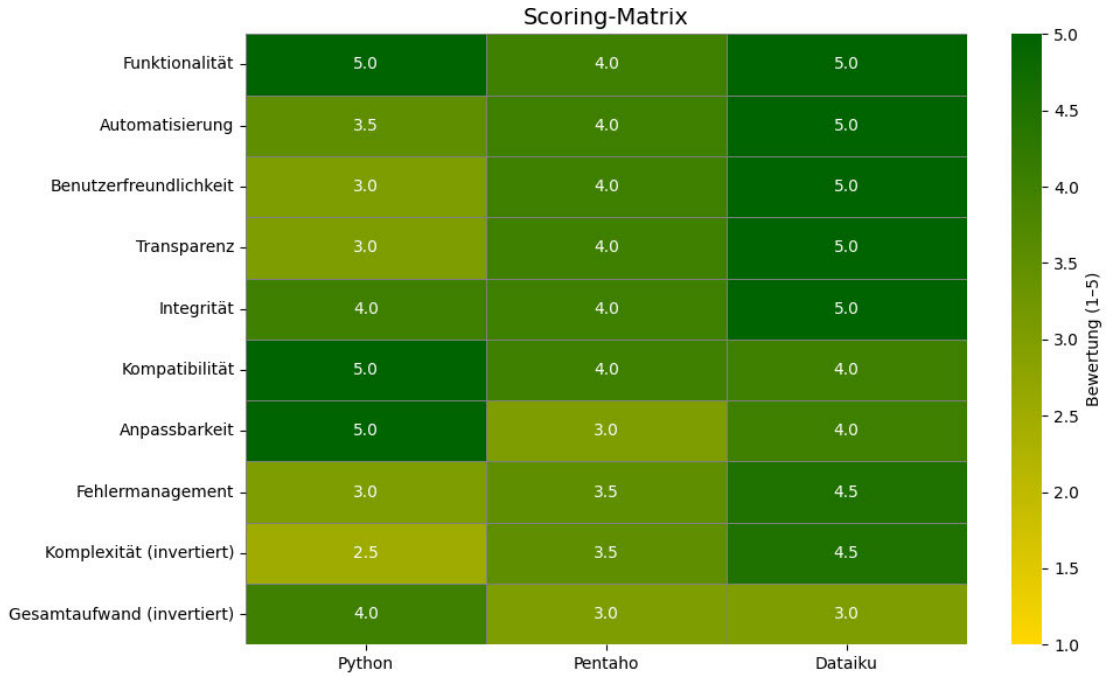


Abbildung 16: Bewertungsmatrix der untersuchten Datenbereinigungstools.

Die **Funktionalität** von Werkzeugen wird anhand der Diversität ihrer Bereinigungsmechanismen beurteilt. Python und Dataiku erzielen in der Bewertung im Vergleich zu Pentaho bessere

Werte von 5,0 bzw. 4,0, unter anderem aufgrund ihrer vorgefertigten Funktionen zur Ausreißerbereinigung. Davon abhängig wird das Maß der **Automatisierung** betrachtet, das für alle drei Tools von 3,5 bis 5,0 inkrementell bewertet wird. Python steht an der niedrigsten Stelle, da es den erheblichsten Aufwand für manuelle Implementierungen erfordert. Daraufhin kommt Pentaho mit seiner interaktiven Benutzeroberfläche, gefolgt von Dataiku, das KI-generierte Module nutzt.

Benutzerfreundlichkeit und **Transparenz** werden im Rahmen eines praktischen Experiments in Verbindung miteinander betrachtet, da beide unter anderem vom Design und der Struktur der Benutzeroberfläche abhängen. Die Bewertung beider Kriterien umfasst die Verständlichkeit, Interaktivität und Visualisierung der umgesetzten Arbeitsprozesse: Dataiku, als visuell ansprechendes Werkzeug, erhält eine Bewertung von 5,0. Aufgrund der möglichen Schwerfälligkeit bei wachsender Komplexität erhält Pentaho eine 4,0, während das Python-Framework mit 3,0 Punkten bewertet wird.

Die **Integrität**, **Kompatibilität** und **Anpassbarkeit** werden direkt durch die technische Softwareschnittstelle beeinflusst. Sie werden beurteilt, basierend auf den Möglichkeiten zur Datenvalidierung, der Unterstützung externer Schnittstellen und Datenformate sowie der Konfigurierbarkeit der Abläufe. Das flexibelste Werkzeug ist Python, das zahlreiche Libraries und funktionale Pakete bietet. Bei den Aspekten Kompatibilität und Anpassbarkeit erhält es die beste Bewertung von 5,0. In der Zwischenzeit wird Pentaho wegen seiner eingeschränkten Funktionalität in diesen Bereichen mit lediglich 3,0 bis 4,0 Punkten bewertet. Man kann Dataiku als einen Python-ähnlichen Ansatz betrachten: Es erzielt 4,0 Punkte im Ranking, dank seiner API-Konnektoren, ML-Plugins und Codeskripte. Was die Integrität angeht, ist die Workflow-Umsetzung in Python (3,5) durch das manuelle Programmieren fehleranfälliger und erfordert eine genauere Steuerung als die vordefinierten Qualitätskontrollmodule wie „Data Validation“ oder „Check Consistency“ in Pentaho (4,0) und Dataiku (5,0).

Die Dimension des **Fehlermanagements** wird gesondert betrachtet, um eine Bewertung der Bereinigungstechniken in Bezug auf ihre Wirksamkeit und Nachvollziehbarkeit bei der Behandlung von Prozessproblemen vorzunehmen. Python erzielt hierbei 3,0 Punkte, weil alle Errors und Exceptions vor der Ausführung berücksichtigt und manuell behandelt werden müssen, abhängig von der Art und Logik des Codes. Pentaho erreicht 3,5 Punkte, weil es durch die

eingebaute Behandlungsfunktion mit anpassbarem Detaillierungsgrad für auftretende Fehler ermöglicht, diese in einer separaten Log-Tabelle zu speichern. Dataiku erleichtert und macht die Fehlererkennung und -bearbeitung visuell nachvollziehbar, indem es erweiterte Anzeigeoptionen, integriertes Arbeitsmonitoring und Protokollierung bereitstellt. Die Log-Ausgaben bei KI-generierten oder komplexen Schritten sind jedoch nicht immer vollständig nachvollziehbar und benötigen oft zusätzliches Kontextwissen, um die Ursachen von Fehlern zu identifizieren. Dies ergibt insgesamt 4,5 Punkte für dieses Tool.

Anschließend werden die **Komplexität** und der **Gesamtaufwand** der Werkzeuge mit invertierter Bewertung verglichen. Dabei werden die Lizenzkosten, der Aufwand für das Setup und die Wartung sowie der Schulungsaufwand für die Einarbeitung berücksichtigt. Aufgrund der notwendigen Kenntnisse zum Programmieren hat Python eine längere und umfassendere Lernkurve, weshalb es in der Komplexitätsbewertung 2,5 Punkte erhält. Auf den nächsten Plätzen sind Pentaho mit 3,5 Punkten und Dataiku mit 4,5 Punkten, die sich hinsichtlich ihrer Arbeitsweise und ihres Automatisierungsgrades unterscheiden. In Bezug auf den Gesamtaufwand erreicht Python 4,0 Punkte, da es als Open-Source-Tool kostenlos und relativ einfach einzurichten ist. Wegen ihrer Lizenzkosten sowie der Notwendigkeit für individuelle Datenflusskonfigurationen und Plugin-Installationen erhalten Pentaho und Dataiku nur 3,0 Punkte.

Weitere Analysen stützen sich auf systemintegriertes Testen sowie Computermessungen der implementierten Bereinigungsalgorithmen, die als End-to-End-Lösungen dargestellt werden.

Der Bearbeitungszeitvergleich zeigt deutliche Unterschiede zwischen den drei Ansätzen: Der programmierbare Prozess in Python hat insgesamt 1,2 Sekunden benötigt, während der Pentaho-Workflow 3 Sekunden und die Dataiku-Lösung etwa 20 Sekunden in Anspruch nahmen (Abbildung 17). Unterschiede bestehen dabei neben der funktionalen und grafischen Ausstattung auch in der Konzeption der einzelnen Algorithmen. Beispielsweise ist der „Left Join“, der bei der Fuzzy-Deduplizierung für den zeilenweisen Vergleich desselben Datensatzes eingesetzt wird, mit einer Komplexität von $O(N^2)$ ressourcenintensiv. Für die optimierte Funktion „rapidfuzz“ in Python fallen die Iterationskosten minimal aus. Der Hauptgrund dafür ist neben der Verarbeitung im Hauptspeicher auch die verbesserte Performance des Algorithmus, die durch Optimierungen der originalen „fuzzywuzzy” mittels Cython und C++ erreicht wurde [35, 37]. Aufgrund der zusätzlichen Analyse der Zwischenergebnisse, der schrittweisen Protokollierung

des Prozesses und der verwendeten Java-Engine benötigt Pentaho bei gleicher lokaler Ausführung mehr Zeit. In Dataiku wird die Bereinigungszeit durch die Zwischenspeicherung der Daten über SQL verzögert.

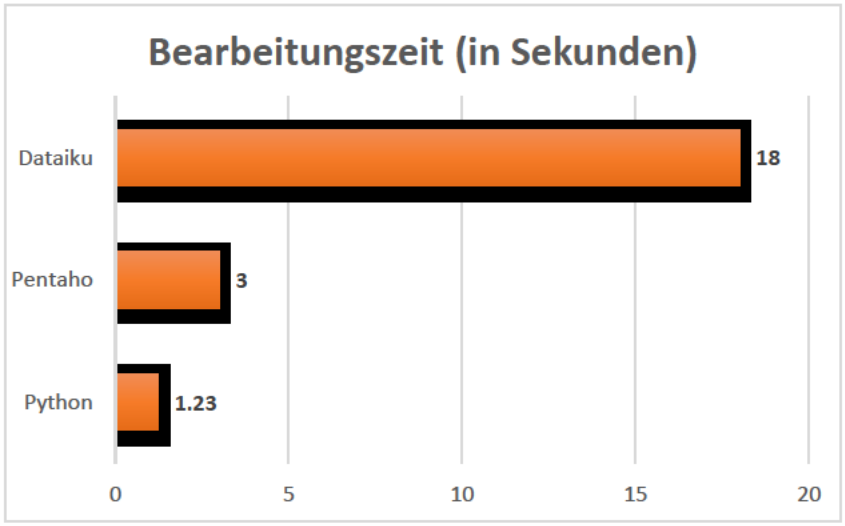


Abbildung 17: Bearbeitungszeit im Vergleich (in Sekunden).

Die durchschnittliche CPU-Auslastung der jeweiligen Bereinigungsprogramme wird im Rahmen der vergleichenden Aufwandsanalyse ebenfalls als technische Kennzahl erfasst. Zum Messen wird der Systemdienst „Ressourcenmonitor“ verwendet. Im Verlauf der Durchführung wird die durchschnittliche Anzahl der CPU-Kerne in Echtzeit erfasst (Abbildung 18).

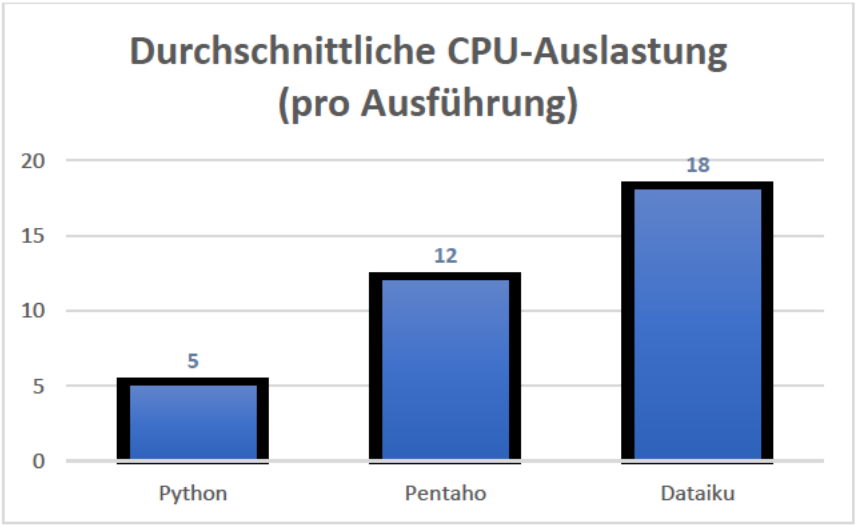


Abbildung 18: Durchschnittliche CPU-Auslastung pro Ausführung.

Die Analyse hat die folgenden Ergebnisse hervorgebracht: Python zeichnet sich durch einen besonders geringen Ressourcenverbrauch aus, da es durchschnittlich nur 5 Nutzungseinheiten benötigt. Pentaho nutzt ungefähr 12 Kerne, weil interne Aufgaben seines Hauptprozesses die CPU-Auslastung erhöhen. Dataiku beansprucht bei lokaler Ausführung bis zu 18 Kerne für die prozessorientierte Parallelisierung und zur Bereitstellung der Plattformdienste im Hintergrund.

Für den qualitativen Vergleich der implementierten Algorithmen werden die resultierenden Datensätze nach jedem Bereinigungsansatz anhand der Anzahl entfernter fehlerhafter Einträge überprüft. Da der saubere Testdatensatz als Basis für die künstliche Verschmutzungsphase dient, ist das Ziel der zu löschenden Inkonsistenzen bereits erkannt. Auf diese Weise lassen sich die Ergebnisse der Erkennungsalgorithmen für Ausreißer und Duplikate in Daten vergleichen (Abbildung 19).

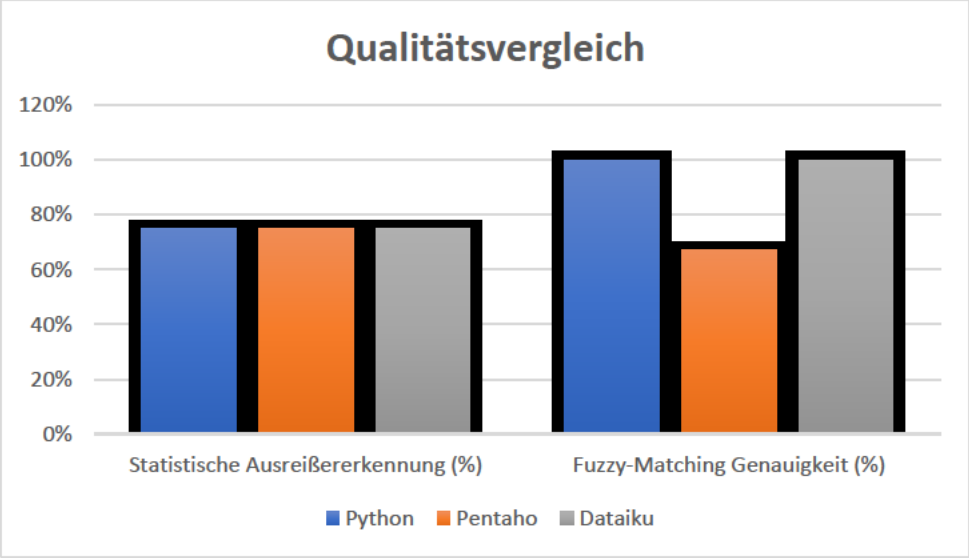


Abbildung 19: Qualitätsvergleich nach Bereinigungsschritten.

Die Bereinigung von Ausreißern erfolgte nach der statistischen IQR-Methode. Dabei erzielten alle drei Verfahren gleiche Ergebnisse: Von insgesamt 4 Ausreißern konnten nur 3 erkannt werden, was anschließend eine zusätzliche bedingte Filterung erforderlich machte.

Für die unscharfe Deduplizierung wurde ein einheitliches Fuzzy-Matching-Konzept angewendet: Mit derselben Logik identifizierte das Pentaho-Modul „Fuzzy match“ 8 von maximal 12 Kandidaten, während die anderen beiden Lösungen eine Trefferquote von 100 % erreichten.

Ein weiterer Schritt in der qualitativen Analyse ist die Evaluierung der unterschiedlichen Strategien zur Befüllung fehlender Werte in komplex zusammenhängenden Attributen. Der Bestimmtheitskoeffizient (R^2) wird verwendet, um den Korrektheitsgrad der prognostizierten Werte in „Customer _Tenure“ zu berechnen. Dieser statistische Messwert liegt zwischen 0 (ungeeignet) und 1 (perfekt). Es wird als Quotient aus der Summe der Quadrate der Residuen des Regressionsmodells (SSRES) und der Gesamtsumme der Quadrate der Abweichungen vom Mittelwert (SSTOT), subtrahiert von 1, definiert:

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_j - \hat{y}_j)^2}{\sum_i (y_j - \bar{y}_j)^2} \qquad [7]$$

Die Analyse beinhaltet die Wiederherstellung der Werte in der Spalte „Customer_Lifetime_Value“ anhand ihres Zusammenhangs mit der bereinigten Spalte „Customer_Tenure“ unter Verwendung eines Random-Forest-Regressionsverfahrens. Danach wird die prozentuale Differenz zwischen den berechneten und den Originalwerten bestimmt. Das heißt: Je genauer „Customer_Tenure“ bereinigt wird, desto deutlicher spiegeln die R^2-Werte diese Genauigkeit wider. Der Prozess wird automatisiert mit den Funktionen der Python-Bibliothek sklearn „RandomForestRegressor“ und „r2_score“ durchgeführt [44].

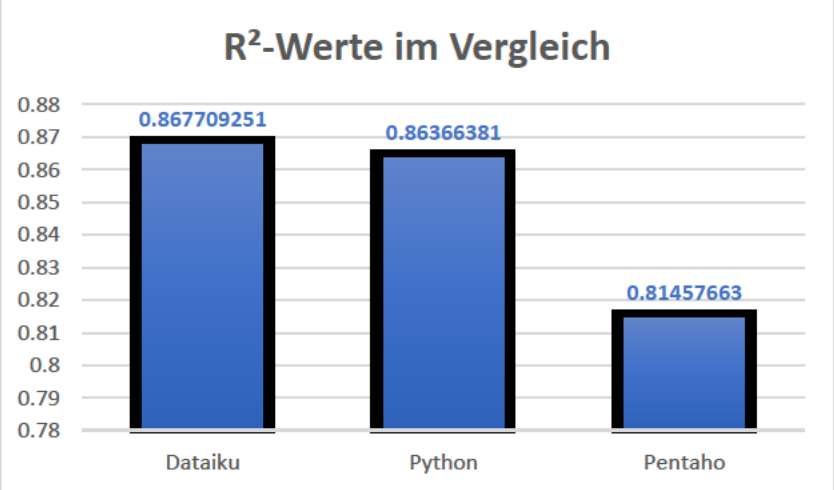


Abbildung 20: Qualitätsvergleich von Imputationstechniken anhand der R^2-Werte.

Die Ergebnisse dieses Testes in Abbildung 20 zeigen den Unterschied zwischen dem Python-kNN-Ansatz, der Pentaho-clusterbasierten Mittelwertimputation und der Extra-Trees-ML-Vorhersage von Dataiku. Im Vergleich zu Python liefert Dataiku die am stärksten

übereinstimmenden Ergebnisse, während Pentaho mit einem um etwa 5 % geringeren Wert schwächer positioniert ist.

Die Resultate dieser Analyse werden im folgenden Kapitel bewertet und in einheitliche Schlussfolgerungen zusammengefasst.

4 Fazit und Diskussion

Im vierten Kapitel wird eine Zusammenfassung der allgemeinen Schlussfolgerungen in Bezug auf das Ziel des Experiments dargestellt, wobei es sich auf die theoretischen Grundlagen des Data Cleansing aus Kapitel 2 und die Analyseergebnisse aus Kapitel 3 stützt. Es gliedert sich in drei Teile: Der Abschnitt „Schlussfolgerungen und Ausblick“ bietet eine Interpretation der Ergebnisse aus dem Unterkapitel 3.3 sowie eine Zusammenfassung der Erkenntnisse der Studie. Im Folgenden werden die im Experiment ermittelten Einschränkungen aufgelistet. Im letzten Teil „Diskussion“ werden offene Fragen für weiterführende Analysen behandelt.

4.1 Schlussfolgerungen und Ausblick

Anhand der im Unterkapitel 3.3 erfassten Forschungsergebnisse lassen sich allgemeine Schlussfolgerungen für alle untersuchten Ansatzeigenschaften ziehen.

Die Bereinigungsansätze zeichnen sich durch technische Merkmale in ihrer Effizienz und durch qualitative Merkmale in ihrer Effektivität aus: Die manuell programmierbare Bereinigung bietet die größte Flexibilität und das beste Ressourcenmanagement, ist jedoch auch die schwerste Option zu bedienen. Obwohl die KI-gesteuerte Datenmanagementplattform die beste Qualität bei höherer Benutzerfreundlichkeit bietet, führt sie zu einer hohen Systemauslastung und kann bei geringen Datenmengen ineffizient sein. Der Standard-ETL-Workflow kann aufgrund begrenzter Funktionalität in der Präzision unterlegen sein, ist dafür aber kostengünstiger und schneller als fortgeschrittene interaktive Methoden.

Die konzeptionelle Evaluierung spielte in der hier durchgeführten Untersuchung eine zentrale Rolle, neben den technischen und qualitativen Merkmalen. Wie verständlich und wartungsfreundlich der Prozess ist, hängt unmittelbar mit dem Interaktionsgrad der Schnittstelle und dem Design der untersuchten Reinigungswerkzeuge zusammen. Dies wird deutlich, wenn das

Datenschema in Echtzeit aktualisiert wird: Bei einem programmierbaren Framework müssen alle Anpassungen an Feldtypen oder Datenstrukturen manuell verfolgt und angepasst werden, vom Validierungsprozess über die Transformation bis zur Ausgabe des endgültigen Ergebnisses. All dies erfordert einen erheblichen Aufwand an manueller Arbeit und Zeit für die Entwicklung, wodurch das Risiko steigt, dass Inkonsistenzen nicht ausreichend behandelt werden. Mit ETL-Integrationstools wie Pentaho ist es möglich, Daten vor ihrer Verarbeitung gezielt zu überprüfen und zu steuern. Diese Werkzeuge verfügen über vorab festgelegte Eingabe- und Zwischenspeichermodule. Obwohl diese Funktionen die Benutzeroberfläche vereinfachen, ist ein robustes System zur Qualitätskontrolle weiterhin notwendig. Intelligente Datenmanagementplattformen wie Dataiku verwenden zentrale Metadatenverwaltung und adaptive Prozessarchitekturen, um Änderungen dynamisch zu erkennen und Datenschemata automatisch anzupassen. Der Verzicht auf starre Eingriffspunkte macht das KI-gesteuerte Bearbeitungssystem trotz höheren Ressourcenverbrauchs besonders geeignet für häufig wechselnde Datenstrukturen und gewährleistet eine nachvollziehbare Wartung und Pflege des gesamten Prozesses. Intelligente Datenmanagementplattformen wie Dataiku nutzen zentrale Metadatenverwaltung und adaptive Prozessarchitekturen, um Änderungen dynamisch zu erkennen und Datenschemata automatisch anzupassen. Der Verzicht auf starre Eingriffspunkte macht das KI-gesteuerte Bearbeitungssystem trotz höheren Ressourcenverbrauchs gut geeignet für häufig wechselnde Datenstrukturen und gewährleistet eine nachvollziehbare Wartung des Prozesses.

In Kapitel 2 der Arbeit wird Data Cleansing als untrennbarer Bestandteil des Wissensgewinnungsprozesses beschrieben, dessen Algorithmen die Zielergebnisse unmittelbar beeinflussen können. Ein praktisches Beispiel hierfür kann die hier beschriebenen Schlussfolgerungen aus dem durchgeführten Experiment bezüglich der Business-Perspektive erweitern.

Angenommen, im bereinigten Abonnentendatensatz sollen die Kunden ermittelt werden, die 80 % des Gesamtkundenwertes ausmachen. Das verwendete Pareto-Prinzip (auch 80/20-Regel genannt) besagt, dass etwa 80 % des Ertrags mit 20 % des Gesamtaufwands erzielt werden können [26]. Zu diesem Zweck werden alle Kundeneinträge zunächst nach dem Kundenwert sortiert. Der resultierende Datensatz wird in 5-%-Cluster von Kunden aufgeteilt und der kumulierte Kundenwert dargestellt. Mithilfe einer Balkendiagrammdarstellung wird anschließend das Verhältnis zwischen den beiden aggregierten Attributen ermittelt. Die Ergebnisse für alle

drei Bereinigungsansätze sind im Anhang D dargestellt und fallen ähnlich aus: 80,14 % der gesamten CLV-Summe werden durch etwa 20,6 % bzw. 20,7 % der gleichen Top-Kunden erzielt. Daher ist der Effekt von qualitativen Bereinigungsunterschieden, wie einer reduzierten Dubletten- und Ausreißerbereinigung in Pentaho, auf bestimmte Mining-Ergebnisse minimal.

Die Ergebnisse des datengesteuerten Geschäftsprozesses können potenziell durch externe Faktoren der Datenbereinigung beeinflusst werden, die nicht direkt mit dem Prozess verbunden sind. Die untersuchten Aspekte lassen sich daher in einer SWOT-Analyse als Stärken, Schwächen, Chancen und Risiken erfassen (Tabelle 3).

Tabelle 3: SWOT-Analyse verwendeter Bereinigungstechniken.

Datenbereinigungsansatz	***Stärken***	***Schwächen***	***Chancen***	***Risiken***
Programmierbare	Hohe Anpassbarkeit, Kompatibilität, Ressourceneffizienz	Hoher manueller Aufwand, schwierige Wartung	Fortgeschrittene Workflows nach individuellen Anforderungen	Prozessinkonsistenz und Fehleranfälligkeit durch manuelle Anpassungen
ETL-basierte	Automatisierbarkeit, Geschwindigkeit, visuelle Unterstützung	Eingeschränkte Funktionalität, sinkende Genauigkeit bei wachsender Logikkomplexität	Erleichterung und Effizienzsteigerung durch Automatisierung	Performance- und Qualitätsverluste bei steigender Workflow-Komplexität
KI-gestützte	Benutzerfreundlichkeit, erweiterte Funktionalität, hohe Genauigkeit	Hoher Systemaufwand, Lizenzkosten	Potenzial zur Vollautomatisierung, kontinuierliche Verbesserung durch lernende Modelle	Datenschutzprobleme, Fehlentscheidungen und Risiken durch Black-Box-Schritte

Die programmierbaren Pipelines mit Python sind besonders flexibel und ermöglichen die Konfiguration einzelner Bereinigungsschritte nach spezifischen Anforderungen bei korrekter Codeumsetzung. Allerdings kann der menschliche Faktor dazu führen, dass Prozessgestaltungen inkonsistent sind oder Fehlerbehandlungen nicht ausreichen.

Das ETL-basierte Vorgehen bietet im Vergleich zu einer reinen Codeimplementierung eine höhere Benutzerfreundlichkeit und Automatisierung, was die Effizienz und Nachvollziehbarkeit der Abläufe verbessert. Allerdings birgt dies Gefahren für die Performance und Qualität, die aus der begrenzten Skalierbarkeit bei komplexeren Dateninkonsistenzen resultieren.

Die intelligente Verarbeitung bietet, dank ihrer funktionalen und architektonischen Vorteile, die Möglichkeit einer kontinuierlichen Optimierung bestehender Abläufe mithilfe von maschinellem Lernen. Im Gegensatz dazu können von KI generierte Schritte die Ausführungszeiten verlängern und aufgrund ihrer Black-Box-Natur Intransparenz sowie Verstöße gegen Datenschutzprinzipien hervorrufen. Dadurch wird die Durchführung von Wartung und Kontrolle erschwert.

Die Wahl des Vorgehens hängt neben den zuvor genannten Aspekten auch wesentlich von der Größe des Problems der Datenverschmutzung, der Struktur der Daten und den Zielsetzungen des Prozesses zur Datenbearbeitung ab. All diese Aspekte spielen in der Datenvorbereitungsphase eine zentrale Rolle und müssen von den jeweiligen Experten berücksichtigt werden.

4.2 Limitationen des Experiments

Im Laufe der experimentellen Analyse treten Beschränkungen auf, die allgemein von der Wahl der Bereinigungstools, dem festgelegten Ziel der Arbeit oder den Testdaten abhängen.

Da die untersuchte Stichprobe automatisiert generiert und in ihrer Größe sowie Anzahl begrenzt wird, werden während des Experiments bestimmte Analyseaspekte unterlassen: Die Reproduzierbarkeit der Ergebnisdaten kann wegen des kontinuierlichen Ablaufs nicht geprüft werden. Die Dateneindeutigkeit sowie die Gesamtintegrität werden außerdem die Untersuchung der Skalierbarkeit der Datenbereinigungsverfahren erschweren. Auch die Zufälligkeit der Fehlerverteilung wird für eine realitätsnahe Abbildung sowie für die Prüfung der Robustheit möglicherweise verletzt.

Die Analyse der untersuchten Bereinigungsanwendungen ist durch folgende Einschränkungen beeinträchtigt: Die Größe des Testdatensatzes ist für eine objektive Messung der Speicherplatzauslastung nicht ausreichend. Der Fokus wird stattdessen auf die Komplexität des Prozesses gelegt. Technische und systembasierte Aspekte wie Kompatibilität, Lizenzdauer und Softwareabhängigkeiten begrenzen dazu auch die verfügbare Funktionalitätsbreite der Anwendungen. Die verwendeten Werkzeuge werden außerdem lokal und ohne zusätzliche Online-Dienste oder Plugins betrieben, um eine Vergleichbarkeit des Experiments zu gewährleisten, sodass die Analyse auf die Standardfunktionalität der Bereinigungskategorie beschränkt bleibt.

Zusammen bestimmen diese Limitierungen den Rahmen dieser Analyse und eröffnen Fragen für weitere Forschungen.

4.3 Diskussion

Der Data-Cleansing-Prozess hat, wie im theoretischen Teil der Arbeit beschrieben, einen stark iterativen Charakter. Bei einem zyklusbasierten Vorgehen erfordert dieser Prozess eine Kontrolle der Bereinigungsqualität durch Fachexperten [19]. Dabei stellt sich die Frage, wie die Rahmenbedingungen für manuelle und vollautomatisierte Eingriffe gesetzt werden sollen, um die optimale Flexibilität und Transparenz des Prozesses sicherzustellen.

Insbesondere bei dem Einsatz automatisierter Verarbeitungsmethoden lassen sich die Standardisierung sowie der Schutz sensibler und geheimer Informationen stärker betonen. Dies ist besonders beim Training von Modellen zur Erstellung möglichst genauer und maßgeschneiderter Bereinigungsalgorithmen der Fall, beispielsweise wenn die Formatierungs- oder Verschlüsselungsprinzipien der Felder des zu verarbeitenden Datenbestands berücksichtigt sein müssen.

Das Ziel eines konzeptionellen Rahmens ist es, die Verarbeitungsprozesse ausreichend und korrekt zu erfassen. Besonders in wiederkehrenden Anwendungsfällen muss das Ablaufloggging für die Nutzer klar und nachvollziehbar sein, um eine reibungslose Wiederverwendbarkeit sicherzustellen. Bei der Integration oder Übertragung des Datenbereinigungssystems in andere Umgebungen sowie bei bereichs- und teamübergreifender Arbeit können Konflikte entstehen. Dies liegt daran, dass die Rollenvergabe und Domänenspezifika nicht nur von der

Kompatibilität, sondern auch von internen Regelungsmechanismen beeinflusst werden. Der daraus entstehende Aufwand kann für die jeweiligen Datenbereinigungstechniken analysiert werden.

Diese Aspekte bieten neue Ansatzpunkte für weiterführende Analysen, die durch bestehende und neue praktische Lösungen sowie Forschungsarbeiten vertieft werden können.

Die methodischen und technischen Aspekte des Data Cleansing werden durch neue Forschungsarbeiten erweitert: In ihrer Studie „Performance and Scalability of Data Cleaning and Preprocessing Tools: A Benchmark on Large Real-World Datasets“ (2025) untersuchen Martins P. et al. die Datenaufbereitungswerkzeuge anhand von Datensätzen, die bis zu 100 Millionen reale, domänenübergreifende Einträge enthalten [30].

Darüber hinaus lassen sich die analysierten Data-Cleansing-Tools durch verschiedene Zusatzpakete, Plugins und Erweiterungen optimieren und spezialisieren. Im Vergleich zu „Pentaho Data Integration“ bietet „Pentaho Data Quality“ von Hitachi eine spezialisierte Funktionalität, während „Pentaho Data Optimiser“ eine Parallelausführung von Bearbeitungsroutinen sowie durch analytische Mechanismen eine kosteneffiziente Datenverwaltung ermöglicht [15, 16].

Data Cleansing bleibt daher ein Bereich, der mit neuen Ansätzen, Werkzeugen und Zielsetzungen weiter untersucht und verbessert werden kann.

Literaturverzeichnis

[1] *Alteryx Inc. Product Overview – Alteryx Data Preparation*. Verfügbar unter: https://help.alteryx.com/dataprep/en/product-overview.html#product-overview [Zugriff am 11. März 2025].

[2] Batini, C. & Scannapieco, M., 2006. *Data Quality: Concepts, Methodologies and Techniques*. Springer.

[3] Batini, C., Cappiello, C., Francalanci, C. & Maurino, A., 2009. *Methodologies for data quality assessment and improvement*. ACM Computing Surveys, 41(3), p. 6–9.

[4] BBC News, 2020. *Excel: Why using Microsoft's tool caused Covid-19 results to be lost*. Verfügbar unter: https://www.bbc.com/news/technology-54423988 [Zugriff am 01. März 2025].

[5] Cleve, J. & Lämmel, U., 2024. *Data Mining: Datenanalyse für Künstliche Intelligenz*. De Gruyter Oldenbourg.

[6] *Dataiku. The Universal AI Platform™*. Verfügbar unter: https://www.dataiku.com/ [Zugriff am 10. Mai 2025].

[7] Dimple, C., Gulati, P., Gupta, T., 2023. *Comparative Study of Missing Value Imputation Techniques on E-Commerce Product Ratings*. Informatica, 47, p. 378.

[8] Ehrlinger, L., Rusz, E. & Wöß, W., 2022. *A survey of data quality measurement and monitoring tools*. Frontiers in big data, 5, 850611.

[9] Gehrau, V., Maubach, K., Fujarski, S., 2022. *Verteilungen*. *In: Einfache Datenauswertung mit R.* Springer VS, Wiesbaden, S., 162, p. 190.

[10] Google Workspace. *Google Sheets: Online Spreadsheets & Templates*. Verfügbar unter: https://workspace.google.com/products/sheets/ [Zugriff am 11. Juni 2025].

[11] Haider, S.N., Zhao, Q. & Meran, B.K., 2020. *Automated data cleaning for data centers: A case study*. Chinese Control Conference (CCC).

[12] Han, J., Kamber, M. & Pei, J., 2011. *Data Mining: Concepts and Techniques*. Elsevier.

[13] Haug, A., Zachariassen, F. & Liempd, D.V., 2011. *The costs of poor data quality*. Journal of Industrial Engineering and Management, 4(2), p. 168–193.

[14] Hitachi Vantara. *Pentaho Data Integration*. Verfügbar unter: https://pentaho.com/products/pentaho-data-integration/ [Zugriff am: 10. Mai 2025].

[15] Hitachi Vantara. *Pentaho Data Optimizer*. Verfügbar unter: https://pentaho.com/products/pentaho-data-optimizer/ [Zugriff am 30. August 2025].

[16] Hitachi Vantara. *Pentaho Data Quality*. Verfügbar unter: https://pentaho.com/products/pentaho-data-quality/ [Zugriff am 30. August 2025].

[17] Hosseinzadeh, M., Azhir, E., Ahmed, O. et al., 2021. *Data cleansing mechanisms and approaches for big data analytics: a systematic study*. Journal of Ambient Intelligence and Humanized Computing, 14, p. 1-13.

[18] Hunt, N., 2002. *Cleaning Dirty Data in Excel*. Teaching Statistics, 24(3), p. 90-92.

[19] Ilyas, I.F., 2016. *Effective Data Cleaning with Continuous Evaluation*. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, p. 41–45.

[20] Ilyas, I.F. & Chu, X., 2019. *Data Cleaning*. ACM Books.

[21] International Organization for Standardization, 2020. *ISO 8000-8:2020 Data Quality – Part 8: Information and Data Quality Management*. ISO.

[22] Jin, Z., 2022. *Principle, Methodology and Application for Data Cleaning techniques*. BCP business & management, 26, p. 731.

[23] Joshi, A.P. & Patel, B.V., 2020. *Data Preprocessing: The Techniques for Preparing Clean and Quality Data for Data Analytics Process.* Oriental Journal of Computer Science and Technology, 13(2–3).

[24] *Kaggle, Kaggle Datasets*. Verfügbar unter: https://www.kaggle.com/ [Zugriff am 08. Mai 2025].

[25] Kimball, R. & Caserta, J., 2004. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wile, p. 167.

[26] Koch, R., 2015. *Das 80/20-Prinzip: Mehr Erfolg mit weniger Aufwand*. Campus Verlag.

[27] Lemon, K. N. & Lemon, L. J., 2010. *Customer Lifetime Value (CLV)*. John Wiley & Sons, Ltd.

[28] Loshin, D., 2010. *Master Data Management*. Morgan Kaufmann.

[29] Maimon, O. & Rokach, L., 2005. *Introduction to Knowledge Discovery in Databases*. Springer.

[30] Martins, P., Cardoso, F., Váz, P., Silva, J., & Abbasi, M., 2025. *Performance and Scalability of Data Cleaning and Preprocessing Tools: A Benchmark on Large Real-World Datasets*. Data, 10(5), p. 68.

[31] *Microsoft. Kalkulationstabellen kostenlos online bearbeiten Microsoft Excel für das Web*. Verfügbar unter: https://excel.cloud.microsoft/ [Zugriff am 11. Juni 2025].

[32] Müller, H. & Freytag, J.C., 2005. *Problems, methods, and challenges in comprehensive data cleansing*. Institut für Informatik, Humboldt-Universität zu Berlin.

[33] Olson, J.E., 2003. *Data Quality: The Accuracy Dimension*. Morgan Kaufmann.

[34] Oni, S., Chen, Z., Hoban, S., & Jademi, O., 2019. *A comparative study of data cleaning tools*. International Journal of Data Warehousing and Mining, 15(4), p. 59–79.

[35] *PyPI. FuzzyWuzzy*. Verfügbar unter: https://pypi.org/project/fuzzywuzzy/ [Zugriff am 27. Mai 2025].

[36] *PyPI. pandas*. Verfügbar unter: https://pypi.org/project/pandas/ [Zugriff am 27. Mai 2025].

[37] *Pypi. RapidFuzz*. Verfügbar unter: https://pypi.org/project/RapidFuzz/ [Zugriff am 27. Mai 2025].

[38] Rahm, E. & Do, H.H., 2000. *Data cleaning: Problems and current approaches*. IEEE Bulletin of the Technical Committee on Data Engineering, 23(4).

[39] Redman, T.C., 2001. *Data Quality: The Field Guide*. Digital Press.

[40] Redman, T.C., 2008. *Data Driven: Profiting from Your Most Important Business Asset*. Harvard Business School Press.

[41] Rohweder, J.P., Kasten, G., Malzahn, D., Piro, A.M.A. & Schmid, J., 2008. *Informationsqualität – Definitionen, Dimensionen und Begriffe. In: Daten- und Informationsqualität*. Vieweg+Teubner, p. 11–44.

[42] Runkler, T.A., 2020. *Data Analytics: Models and Algorithms for Intelligent Data Analysis*. Springer Fachmedien, p. 2.

[43] *Scikit-learn. sklearn.impute.KNNImputer — scikit-learn 1.5.2 documentation*. Verfügbar unter: https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html [Zugriff am 5. Juni 2025].

[44] *Scikit-learn. sklearn.metrics.r2_score — scikit-learn 1.5.2 documentation*. Verfügbar unter: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html [Zugriff am 8. Juli 2025].

[45] García, S., Luengo, J., Herrera, F., 2015. *Data preprocessing in data mining*. Cham, Switzerland: Springer International Publishing, p. 45.

[46] SimpleML for Sheets. *SimpleML for Sheets Documentation*. Verfügbar unter: https://simplemlforsheets.com/ [Zugriff am 11. Juni 2025].

[47] Singh, S.K. and Dwivedi, D.R.K., 2020. *Data mining: dirty data and data cleaning*. Available at SSRN 3610772.

[48] Statista, 2024. *Most used programming languages among developers worldwide, as of 2024*. Verfügbar unter: https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/ [Zugriff am 10. Juni 2025].

[49] *Visual Studio Code. Data Wrangler*. Verfügbar unter: https://code.visualstudio.com/docs/datascience/data-wrangler [Zugriff am 12. Mai 2025].

[50] Wang, R.Y. & Strong, D.M., 1996. *Beyond Accuracy: What Data Quality Means to Data Consumers*. Journal of Management Information Systems, 12(4), p. 18–23.

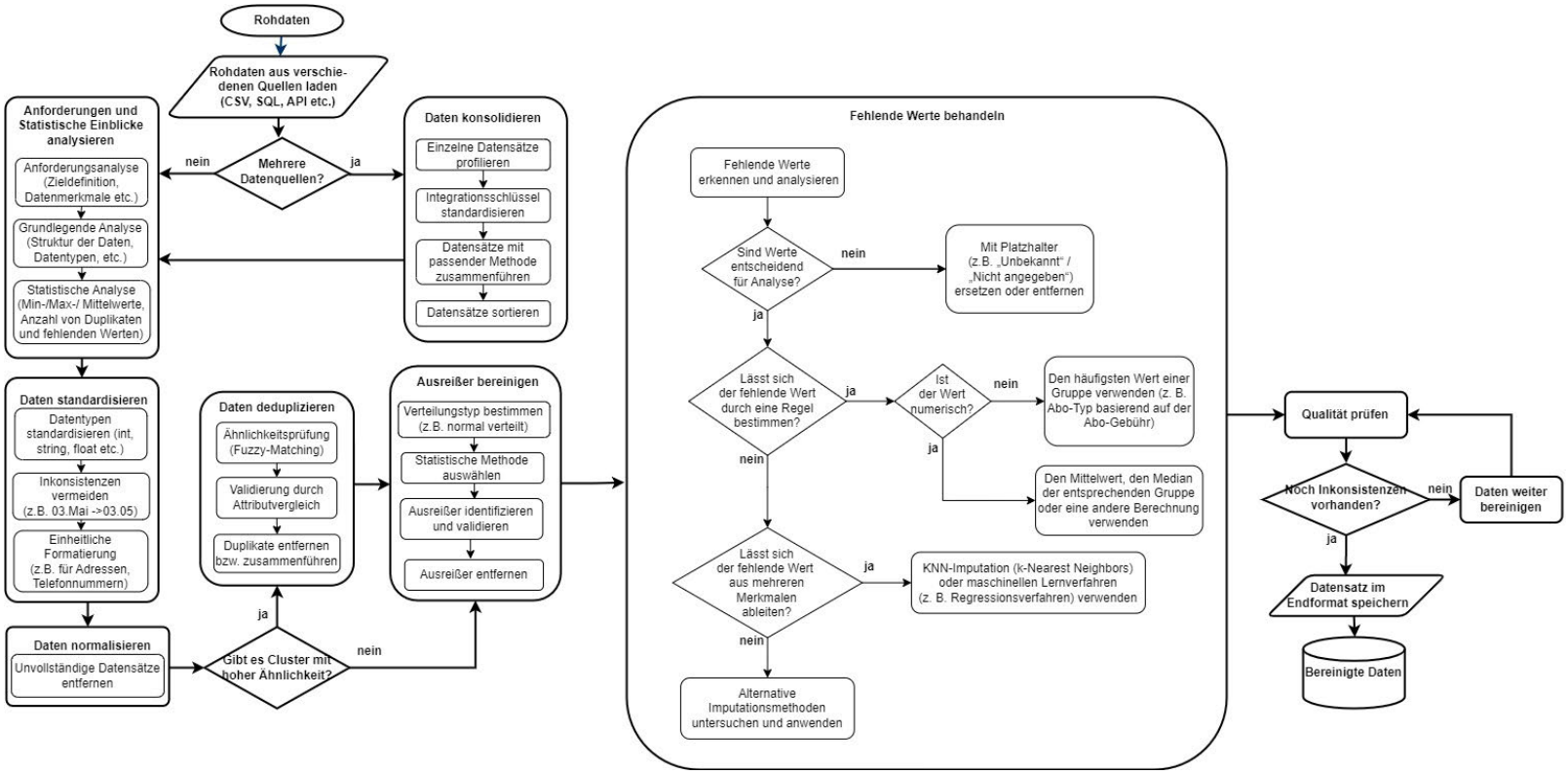
A Anhang 1

Vergleichender Überblick über Datenbereinigungstools.

Kategorie	*Beschreibung*	*Typische Merkmale*	*Vorteile*	*Nachteile*	*Beispiele*
Tabellenbasierte manuelle Tools	Interaktive Datenverarbeitung direkt mittels Tabellenkalkulation	Filter, Formeln, bedingte Formatierung, Pivot-Tabellen	Intuitive Bedienung, ausgezeichnete Nachvollziehbarkeit, keine Programmierkenntnisse nötig	Geringer Automatisierungsgrad, limitierte Bearbeitungskapazitäten, Funktionseinschränkungen	Microsoft Excel, Google Sheets
Individuell programmierte Lösungen	Skriptbasiertes, spezialisiertes Datenmanagement	Funktionssteuerung über Bibliotheken, reibungslose Einbindung in Arbeitsabläufe	Stark anpassbar, frei verfügbar (Open Source)	Technische Kenntnisse erforderlich, erhöhter Wartungsaufwand bei unzureichender Dokumentation	Python (Pandas), R (dplyr, tidyr), SQL
ETL- und Datenintegrationsplattformen	Visualisierung von Datenprozessen und deren schrittweise Optimierung	Drag-and-Drop-Workflows, Integration mehrerer Datenquellen, Metadatenmanagement	Automatisierung wiederholbarer Aufgaben, Skalierbarkeit, Datenkonsistenz und -integrität	Komplexer Prozessaufbau, kostenintensiv, wenig für spontane Analysen geeignet	Talend Open Studio, Pentaho Data Integration, Alteryx Designer
Cloudbasierte KI- und ML-Technologien	Einsatz von KI-Agenten und intelligenten Algorithmen für die Datenbereinigung	Mustererkennung, No-/Low-Code-Workflow, adaptive Optimierung durch kontinuierliches Lernen	Vollautomatisiert, skalierbar, benutzerfreundlich	Potenziell hoher Ressourcen- und Kostenbedarf, Datenschutz- und Compliance-Risiken	Google Cloud Dataprep, Dataiku, Zoho DataPrep

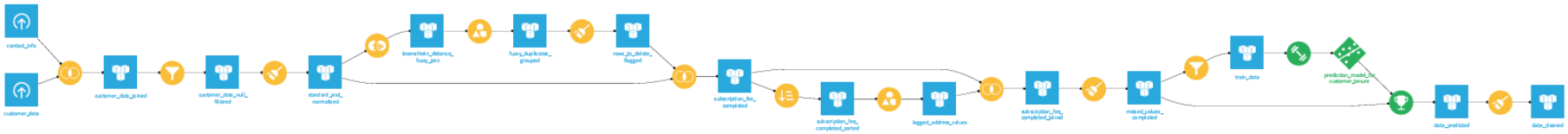
B Anhang 2

Ablaufdiagramm des implementierten Datenbereinigungsprozesses



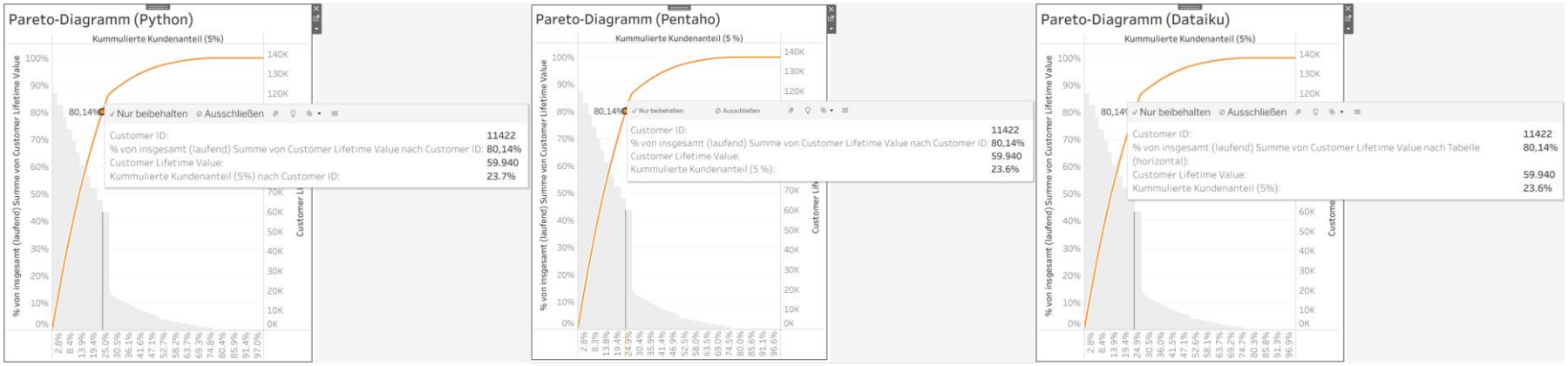
C Anhang 3

Workflow zur Datenbereinigung in Data DSS



D Anhang 4

Überprüfung der Kundenwertverteilung nach dem Pareto-Prinzip.



Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Hamburg	02.10.2025	[redacted]
Ort	Datum	Unterschrift im Original