technische universität dortmund

IT in Produktion und Logistik

Masterarbeit

Anwendung von Data Mining auf produktionslogistischen Massendaten mit Schwerpunkt Datenvorverarbeitung

verfasst von

Yanjun Li

Matrikel-Nr.: 169889

Studiengang: Logistik

Ausgegeben am: 04.07.2016

Eingereicht am: 19.12.2016

Betreuer:

Univ.-Prof. Dr.-Ing. Markus Rabe

Dipl.-Inf. Anne Antonia Scheidler

Technische Universität Dortmund
Fakultät Maschinenbau
Lehrstuhl für IT in Produktion und Logistik

Inhaltverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Formelsverzeichnis

Abkürzungsverzeichnis

CL	Cluster
DM	Data Mining
EM	Erwartungsmaximierung
ERM	Entity-Relationship-Modellierung
FDT	“FAIL”-Datentabelle
FS	Feature Selection
GUI	Graphical User Interface
HDT	Hauptdatentabelle mit 100.000 Datenzeilen
hFre	High frequency
HRDT	“High Repairs”-Datentabelle
KDD	Knowledge Discovery in Databases
lFre	Low frequency
LRDT	“low repairs”-Datentabelle
ManuTime	ManufacturingTime
MESC	Musterextraktion in SCs
mFre	Middle frequency
mid.	middle
More7RDT	“NmbOfRepairs more than 7”-Datentabelle
n. b.	nicht bestimmbar
NE	Nicht erscheinen
ParDesId	ParameterDescriptionId
RH	Relative Häufigkeit
RouSe	RoutingSequence
RS	ResultSequence

SC	Supply Chain
UFS	Unsupervised Feature Selection
WPId	WorkPlaceId
WS	WorkSequence

1 Einleitung

Data Mining bedeutet die Auswahl, Reinigung, Verarbeitung, Analyse und Extraktion von nutzungsvollen Erkenntnissen aus den Rohdaten [Agg15, S. 1]. Daten spielen eine zentrale Rolle in der Informationstechnologie, wobei mit ihrer Hilfe für das Unternehmen notwendigen Informationen übermittelt werden [Pet05, S. 1]. In einem Unternehmen werden Daten in zahlreichen Bereichen angewendet, z. B. industrielle Prozessdaten, Geschäftsdaten, Textdaten und strukturierte Daten [Run15, S. 1f.]. Die Daten in Unternehmen werden entweder von automatisierten oder von nicht automatisierten Prozessen erzeugt und verarbeitet. Während der beiden Prozessarten werden immer neue Daten erstellt, und somit wächst die Datenmenge eines Unternehmens, sind es dann Massendaten [Pet05, S. 1]. In der unternehmerischen Praxis spielt die Auswertung von umfangreichen Massendaten eine wichtige Rolle für die Entscheidungen eines Unternehmens, zum Beispiel sind eine Erhöhung der Lieferbereitschaft und eine Optimierung des Lagerbestands zu unterstützen [Leh16, S. 190]. Traditionell wird die Datenbeschaffung als eine der wichtigsten Phasen der Datenanalyse betrachtet. Der Analyst benutzt hierfür sein verfügbares fachliches Wissen zur Auswahl der Daten, die gesammelt werden sollen. Bei diesem Fall ist die Summe der ausgewählten Daten normalerweise beschränkt, damit der Datenbeschaffungsprozess manuell durchgeführt werden kann [RM15, S. 2f.]. Für die Massendaten sind die traditionellen Datenanalyse-Verfahren ineffizient.

Damit stellt sich die Frage: Wie sollen die Erkenntnisse und Muster innerhalb der Massendaten des Unternehmens extrahiert werden? Die Extraktion von komplexen Mustern ist eine wichtige Voraussetzung zur Zielerreichung, damit die Daten besser erfasst und überblickt werden können und das Wissen von den Massendaten extrahiert werden kann [Leh16, S. 190]. Wegen der hohen Anzahl an Daten funktionieren die traditionellen manuellen und statistischen Bearbeitungsverfahren nicht mehr, um das Muster innerhalb der Massendaten herauszufinden. Oftmals werden daher heutzutage Data Mining (DM)-Verfahren eingesetzt zur Extraktion von Mustern innerhalb der Massendaten und zur Einteilung der Massendaten in bestimmte Gruppen [CL16, S. 2]. Der Einsatz von DM-Verfahren ist ein Schritt des KDD-Prozesses (Knowledge Discovery in Data Bases). Die Hauptaufgaben davon sind die Anwendung der Datenanalyse und das Herausfinden eines Algorithmus, eines besonderen Rechnungsverfahrens, mit dem ein spezielles Muster mithilfe der meisten vorhandenen Kapazitätsbegrenzung des Computers extrahiert werden kann [FPS96, S. 41]. In der Realität sind Daten normalerweise von Fehlern und Rauschen begleitet. Deshalb ist die Durchführung der Datenvorverarbeitung notwendig [Run15, S. 23].

Der Schwerpunkt dieser Arbeit besteht aus zwei Hauptaspekten. Der erste Schwerpunkt ist die Untersuchung von Methoden der Datenvorverarbeitung bezüglich der Massendaten und deren Anwendung auf die Firmendaten. Der zweite Schwerpunkt ist die Extraktion von versteckten Clustern innerhalb der Firmendaten mithilfe eines Clusteranalyseverfahrens.

Die Experimentdaten dieser Masterarbeit wurden von einem produktionslogistischen Unternehmen gesammelt. Dazu werden die folgenden fünf Aufgaben gestellt: Die erste Aufgabe konzentriert sich auf die Recherche der Theorien von Datenvorverarbeitungs-Methoden, nämlich Datenhomogenisierung, Datenaggregation und Feature Selection. Als nächste Aufgabe soll ein Zielformat für die relevanten Datenbestände der Experimentdaten entwickelt werden. Die dritte

Aufgabe betrifft die Identifikation der Erweiterungspotenziale der Datentabellen nach den konkreten Data Mining-Fragestellungen. Nach der Vorbereitung der theoretischen Kenntnisse sollen als die vierte Aufgabe eine spezifische Fragestellung ausgewählt und der Datenvorverarbeitungsprozess durchgeführt werden, wobei die Auswahl der geeigneten Datenaggregationsstufe und der Feature Selection-Prozess nach der spezifischen Fragestellung durchgeführt werden. Nach dem Vorverarbeitungsprozess wird dann der DM-Prozess nach der ausgewählten Fragestellung vorgenommen. Die Fragestellung dieser Masterarbeit wurde zuvor als der zweite Schwerpunkt dieser Masterarbeit erklärt. Der DM-Prozess in dieser Masterarbeit wird nach dem MESC-Vorgehensmodell durchgeführt, das am ITPL entwickelt und als Whitepaper zur Verfügung gestellt wurde. Zum Schluss soll die praktische Verwertbarkeit des Vorgehensmodells untersucht werden.

Diese Arbeit wird nach den folgenden Gedanken aufgebaut. Im theoretischen Teil werden am Anfang die Grundbegriffe wie Daten und der KDD-Prozess vorgestellt. Danach werden die möglichen Probleme von Data Mining aufgezählt und es wird die Notwendigkeit der Datenvorverarbeitung herausgearbeitet. Anschließend werden die Datenvorverarbeitungsschritte in folgenden Methoden genau beschrieben: Datenhomogenisierung, Datenaggregation und Feature Selection. Dann wird die Clusteranalyse nach dem Bedarf der Fragestellung genau erläutert.

Im Praxisteil werden zuerst die Vorbereitungsschritte der Modellierung dargelegt, nämlich die Aufgabendefinition am Anfang, die Auswahl und Integration der relevanten Datenbestände, die Erstellung des Datenmodells durch ERM (Entity Relation Modelling) und eine kurze Vorstellung der angewendeten Software „RapidMiner“. Nach der Erläuterung der Experimentvorbereitung wird der Datenvorverarbeitungsprozess nach den obengenannten drei Schritten ausführlich mit der jeweiligen Darstellung der Experimentdaten und des zugehörigen Modells erläutert. Nach dem Ergebnis der Datenanalyse und entsprechend der Fragestellung werden die geeignete Datenaggregationsstufe und die Attribute im Abschnitt Datenvorverarbeitung festgelegt und ausgewählt. Als Ergebnis der Datenvorverarbeitung werden die originalen Firmendaten nach der Aufgabenstellung vorverarbeitet und sind dann geeignet für die spätere Clusteranalyse. Die möglicherweise produktionsrelevanten Unregelmäßigkeiten, die durch die Datenanalyse aus den Firmendaten extrahiert werden, werden mithilfe einer Vergleichstabelle angezeigt.

Durch die Datenanalyse, die Recherche nach unterschiedlichen Clusteralgorithmen und die zahlreichen Experimente im RapidMiner werden zwei geeignete Algorithmen für die vorliegenden Firmendaten ausgewählt, um die Daten zu gruppieren und die versteckten Cluster zu entdecken. Der Experimentprozess wird mithilfe von Screenshots und schriftlicher Interpretation dargestellt und nachvollziehbar gemacht. Anschließend werden die beiden Algorithmen nach geeigneten Methoden evaluiert und die Ergebnisse der beiden Algorithmen werden verglichen. Dann werden die Clusteranalyse-Ergebnisse mithilfe schriftlicher Erläuterung, visueller Grafiken und Tabellen nach jedem einzelnen Clusteralgorithmus interpretiert. Zum Schluss werden die Ergebnisse der beiden Algorithmen zusammengefasst und die nützlichen Kenntnisse werden extrahiert.

Nach dem experimentellen Teil der Arbeit wird die praktische Verwertbarkeit des Vorgehensmodells nach der konkreten Durchführung entsprechend der zuvor ausgewählten Fragestellung aufgezeigt und die daraus eventuell noch existierenden Probleme werden vorgestellt. Gleichzeitig werden entsprechende Verbesserungsmethoden vorgeschlagen. Ein modifiziertes

Vorgehensmodell zur Datenvorverarbeitung wird im Kapitel 5 als ein wissenschaftlicher Beitrag vorgestellt und erläutert.

Zum Schluss wird die ganze Arbeit zusammengefasst und es wird ein Ausblick über die eventuelle Weiterarbeitungsrichtung dieses Themas gegeben.

Alle Experimentmodelle, die in der Software RapidMiner zur Durchführung der Datenvorverarbeitung und Clusteranalyse aufgebaut werden, werden exportiert und in der begleitenden CD gespeichert.

2 Data Mining als Teil des KDD-Prozesses

Data Mining (DM) bedeutet die Anwendung von speziellen Algorithmen auf Daten zur Extraktion von Mustern und wird heutzutage häufig in der Statistik, zur Datenanalyse und für das Management der Informationssystem-Kommunikation verwendet. DM ist ein Schritt des KDD-Prozesses und besteht aus zwei Hauptaspekten, nämlich der Datenanalyse und der Erkennung von geeigneten Algorithmen zur Extraktion von nutzbaren Mustern, die sich innerhalb der Daten verbergen. Die Schwerpunkte dieser Masterarbeit sind die Datenvorverarbeitung und die Clusteranalyse. In diesem Kapitel werden die theoretischen Grundlagen erläutert, die für die späteren Experimente relevant sind. Zuerst werden die Grundbegriffe und das Vorgehensmodell vorgestellt. Anschließend werden die drei Hauptschritte der Datenvorverarbeitung behandelt. Zum Schluss wird die Clusteranalyse mit verschiedenen Aspekten genau erläutert.

2.1 Daten und KDD-Prozess

Am Anfang dieses Kapitels werden das Basisobjekt des DM-Prozesses „Daten“ und der dem Data Mining übergeordnete Begriff „KDD-Prozess“ kurz erläutert.

2.1.1 Daten und Attribute

Daten sind die Basiseinheit des DM-Prozesses sowie des KDD-Prozesses. In diesem Abschnitt werden die Grundbegriffe von Daten und Attributen sowie ihre Haupttypen erläutert.

Daten

Nach [DIN95] wird der Fachbegriff „Daten“ so definiert:

„Daten sind Zeichen oder kontinuierliche Funktionen, die aufgrund von bekannten oder unterstellten Abmachungen und zum Zweck der Verarbeitung Informationen darstellen.“

In einem Unternehmen wird die Datenverarbeitung als ein wichtiger Teil des Geschäftsprozesses betrachtet [KD15, S. 21]. Die Daten-Wissenschaft ist die Disziplin der Verarbeitung und Analyse von Daten, um wertvolle Kenntnisse aus Daten zu extrahieren [RM15, S. 1].
In dieser Masterarbeit werden viele Fachbegriffe angewendet, die sich auf das Thema „Daten“ beziehen. Zur Vermeidung der Vermischung werden entsprechende Begriffe nun kurz erläutert.

Datenbank
Eine Datenbank funktioniert als ein Datenverwaltungssystem, damit die Massendaten ständig und ohne Widerspruch gespeichert werden können. Mithilfe der Datenbank können die Daten jederzeit nach dem Bedarf exportiert werden [OGT04, S. 31].

Datenbestand
Der Begriff „Datenbestand“ wird als die Gesamtmenge der Daten bezeichnet, die in der Datenbank und in der Datenverarbeitung gespeichert werden [BW73, S. 75].

Dataset

Dataset bedeutet jede organisierende Sammlung von Daten. Ein Dataset kann in Form einer Datentabelle, eines Textabschnitts oder einer Webseite vorliegen [Oec16].

Datensatz & Datenfeld

Der Datenbestand einer Datenbank wird in mehrere kleine Teile untergliedert, nämlich die Datensätze (oder Datenzeilen). Ein Datensatz selbst kann noch in mehrere Einheiten unterteilt werden, nämlich Datenfelder. Normalerweise ist die Struktur aller Datensätze innerhalb einer Datentabelle gleich. Der Eintrag, der durch einen Datensatz und eine Spalte der Datentabelle zugeordnet wird, wird als ein Datenfeld betrachtet [GA13, S. 375].

Etikettierte und *Nicht etikettierte Daten*
Diese zwei Datentypen werden auf English *labelled* und *unlabelled Data* genannt. Die etikettierten Daten haben speziell bezeichnete Attribute und das Ziel ist die Prognose des Wertes der Attribute für die Zukunft mit den gegebenen Daten. Der DM-Prozess mit diesem Datentyp wird *supervised learning* genannt, beispielsweise in Form von Klassifikationsverfahren. Im Gegensatz dazu werden die Daten, die keine speziell bezeichneten Attribute haben, als nicht etikettierte Daten bezeichnet. Der DM-Prozess von diesem Datentyp wird *unsupervised learning* genannt, wie beispielsweise die Clusteranalyse [Bra07, S. 4].

Attribute
Nach [WFH11] wird der Begriff „Attribut“ wie folgt definiert: Daten werden mithilfe der Werte in einer festen und vorher definierten Menge an Attributen oder Features beschrieben. Die Datensätze stehen in den Zeilen der Datentabellen, während die Attribute die Spalten der Datentabellen sind. Die konkreten Werte eines Attributes werden auch „Attributwert“ genannt.

Grundsätzlich wird der Begriff „Attribut“ in zwei Typen unterteilt, nämlich numerisches und nominales Attribut. Das numerische Attribut wird manchmal auch „kontinuierliches Attribut“ genannt. Mithilfe dieses Attributs werden Nummern-Werte gemessen, die entweder eine reelle Zahl oder eine ganze Zahl sind. Im Gegensatz dazu werden die vorher definierten Werte als endliche Menge von Möglichkeiten mithilfe des Attributtyps „nominales Attribut“ beschrieben. Das nominale Attribut wird auch „kategorisches Attribut“ genannt. In der Statistik gibt es noch andere Attributtypen, wie beispielsweise das Original-Attribut und das Ratio-Attribut, um das Messniveau festzulegen [WFH11, S49].

2.1.2 Übersicht des KDD-Prozesses

In der Realität werden Daten aus verschiedenen Bereichen mit einer dramatischen Geschwindigkeit gesammelt und akkumuliert. Eine neue rechnerische Theorie und Werkzeuge sind notwendig, um den Menschen bei der Extraktion der nützlichen Informationen aus der schnell wachsenden Menge an digitalen Daten zu helfen. Solche neuen Theorien und Werkzeuge sind das Hauptthema des KDD-Prozesses. In der abstrakten Ebene beschäftigt sich der KDD-Prozess mit der Entwicklung der Methoden und Techniken, um die Daten erläutern zu können. Der Kern des KDD-Prozesses ist die Anwendung der speziellen Data Mining-Verfahren zur Herausbildung und Extraktion der potenziellen Muster der Originaldaten [FPS96, S. 37]. Nachfolgend

wird eine Übersicht der Schritte des KDD-Prozesses nach [FPS96, S. 41] mithilfe der Abbildung 2.1 gegeben.

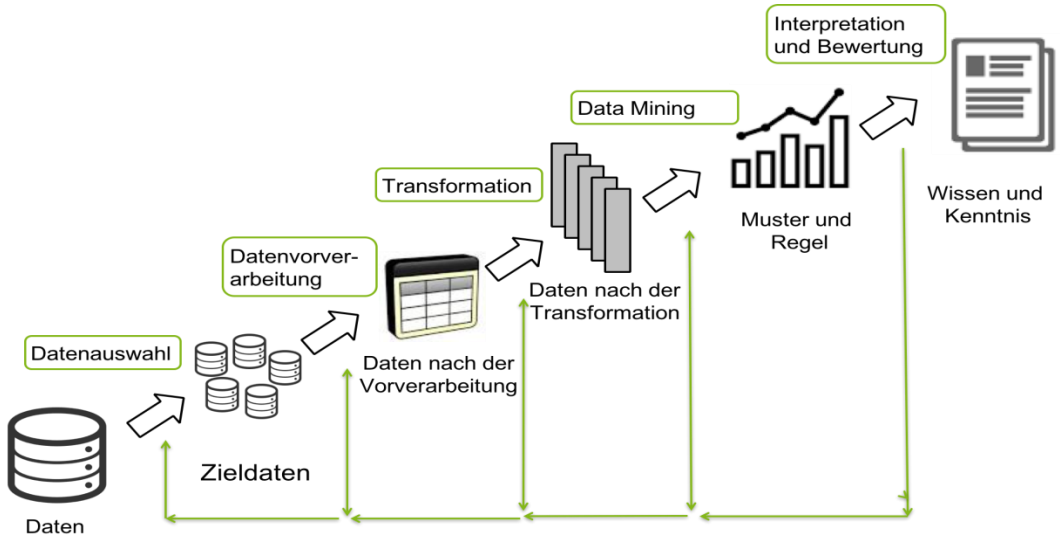


Abbildung 2.1: Übersicht des KDD-Prozesses (nach [FPS96, S. 43]).

Nach der Sammlung der Originaldaten wird zuerst ein Verständnis der Daten entwickelt, wie beispielsweise die Anwendungsdomäne der Daten, damit das Ziel des KDD-Prozesses abgeleitet werden kann. Nach den Bedürfnissen der praktischen Datenanalyse wird ein Teil der Originaldaten ausgewählt. Danach kommt der Schwerpunkt dieser Masterarbeit zum Tragen: die Datenvorverarbeitung. Die vorher ausgewählten Daten werden in diesem Schritt vorverarbeitet, um sie sauber und problemlos für den späteren DM-Prozess bereitzustellen. Bevor der DM-Prozess durchgeführt wird, werden die vorverarbeiteten Daten zur speziellen Repräsentation transformiert, die für eine bestimmte DM-Methode geeignet ist. Entsprechend dem Ziel des KDD-Prozesses wird die geeignete DM-Methode ausgewählt und durchgeführt. Dann werden die benötigten Parameter und geeignete Algorithmen festgelegt, damit das Modell aufgebaut werden kann. Nach der erfolgreichen Durchführung des DM-Prozesses werden die versteckten Muster innerhalb der Originaldaten gesucht, um das Wissen und die Kenntnisse aus den Originaldaten zu extrahieren. Zum Schluss werden die entdeckten Muster interpretiert und evaluiert, womit die gesuchten Kenntnisse erhalten werden [Sha13, S. 61].

Es wird zusammengefasst, dass Originaldaten in der Realität normalerweise viele unterschiedliche Probleme aufweisen. Viele Werte fehlen und sind inkonsistent durch unterschiedliche Datenquellen. Solche Probleme verhindern die effiziente Datenanalyse. Deshalb ist die Datenvorverarbeitung vor dem DM-Prozess notwendig [Agg15, S. 27].

2.2 Übersicht über den Data Mining-Prozess

Bevor die genauen Schritte der Datenvorverarbeitung erläutert werden, wird eine Übersicht des kompletten DM-Prozesses nach dem Whitepaper des ITPL vorgestellt und es werden die konkreten Phasen des DM-Prozesses, die einzelnen Schritte dabei und eine entsprechende Kurzbeschreibung eingeführt.

2.2.1 Data Mining-Vorgehensmodell des ITPL

Nach [ITP16] wird in der Tabelle 2.1 das MESC-Vorgehensmodell dargestellt, die eine Transformation vom KDD-Prozess zum Supply Chain-Bereich ist. Die Struktur im Modellierungsabnitt dieser Masterarbeit wird grundsätzlich nach diesem Vorgehensmodell aufgebaut.

Tabelle 2.1: Vorgehensmodell zur Musterextraktion in SCs (MESC) (nach [ITP])

Phase	Schritte	Kurzbeschreibung
1. Aufgabendefinition	1.1 Bestimmung der Aufgabenstellung	Formulierung der Aufgabenstellung des Supply Chain Managements (SCM) unter Berücksichtigung von gegebenen Randbedingungen und Festlegung der Zielkriterien
2. Auswahl der relevanten Datenbestände	2.1 Datenbeschaffung	Bestimmung und Zugang zu den Datenquellen und den zugehörigen Datenbeständen gemäß Zieldefinition
	2.2 Datenauswahl	Auswahl der Datenbestände mittels Kontextwissen (für Def. siehe Bullinger et al. 2009) zwecks Datenreduktion
3. Datenvorverarbeitung	3.1 Formatstandardisierung	Überführung der selektierten Datenbeständen in ein Standardformat
	3.2 Gruppierung	Fachliche Gruppierung der Datenbestände unter Berücksichtigung der Aufgabenstellung
	3.3 Datenanreicherung	Datenanreicherung unter Einbeziehung von Kontextwissen
	3.4 Transformation	Prüfung auf Atomarität der Attribute, Anreicherung von Daten unter Zuhilfenahme von Kontextwissen, Merkmalsreduktion, Behandlung von fehlenden und fehlerhaften Merkmalen sowie Ausreißerkorrektur
4. Vorbereitung des Data-Mining- Verfahrens	4.1 Verfahrenswahl	Auswahl des einzusetzenden Verfahrens in Abhängigkeit zur Aufgabenstellung
	4.2 Werkzeugauswahl	Auswahl eines geeigneten Data-Mining-Werkzeug
	4.3 Fachliche Kodierung	Fachliche Auswahl und Kodierung geeigneter Attribute
	4.4 Technische Kodierung	Technische Auswahl und Kodierung geeigneter Attribute
5. Anwendung der Data-Mining-Verfahren	5.1 Entwicklung eines Data-Mining-Modells	Modellentwicklung und Trennung der Datenbestände in Trainings-, Validierungs- und Testdaten
	5.2 Training des Data-Mining-Modells	Training des Data-Mining-Modells mittels Validierung aus 5.1
6. Weiterverarbeitung der Data-Mining-Ergebnisse	6.1 Extraktion handlungsrelevanter Data-Mining-Ergebnisse	Unter Berücksichtigung der Handlungsrelevanz sowie technischen Maßzahlen sind für das SCM interessante Ergebnisse

		zu extrahieren.
	6.2 Darstellungs-transformation der Data-Mining-Ergebnisse	In Abhängigkeit der eingesetzten Data-Mining-Verfahren sowie der Aufgaben-stellung müssen die Ergebnisse in eine explizite Darstellungsform überführt werden

2.2.2 Data Mining-Aufgaben

Generell sind die Aufgaben von Data Mining nach [TSK06, S. 7] in zwei hauptsächliche Kategorien unterteilt.

Prognose-Aufgabe
Das Ziel dieser Aufgabe ist die Prognose von Werten anhand von speziellen Attributen, die auf den Werten von anderen Attributen basieren. Die Attribute, die prognostiziert werden sollen, werden normalerweise als die Ziel- oder abhängige Variable betrachtet, während die Attribute, die für die Durchführung der Prognosen angewendet werden, als erklärende oder unabhängige Variablen angesehen werden.

Beschreibungsaufgabe
Die Ziele von Data Mining in diesem Fall sind, die Schemata von Korrelation, Cluster und Ausreißer herauszufinden, die die vorliegenden Zusammenhänge innerhalb der Daten erfassen. Die Beschreibungsaufgaben vom DM-Verfahren beziehen sich oftmals auf die Untersuchung der Natur und eine Nachbearbeitungstechnik ist häufig notwendig, um das Ergebnis zu validieren und zu erläutern. In dieser Masterarbeit wird das DM-Verfahren sich mit dieser Aufgabe beschäftigt, um die Firmendaten zu analysieren und die nützlichen Kenntnisse dabei herauszufinden.

Die DM-Techniken unterteilen sich grundsätzlich in zwei Bereiche, nämlich die statistischen Verfahren und die Maschinen-Lernen-Verfahren. Die Unterschiede zwischen diesen zwei Verfahren sind, dass sich statistische Verfahren besonders für die Analyse von kleineren und vorstrukturierten Datenmengen eignen. Im Gegensatz dazu sind umfangreiche und schlecht strukturierte Daten mehr geeignet für das Maschinen-Lernen-Verfahren [Blu06, S. 28].

2.3 Datenvorverarbeitung

Die Phase der Datenvorverarbeitung ist wahrscheinlich der wichtigste Bestandteil des DM-Prozesses. Eigentlich sollte sie stark beachtet werden, jedoch wird sie in der Praxis kaum als wichtig berücksichtigt, da normalerweise mehr Wert auf den späteren DM-Prozess gelegt wird [Agg15, S. 5]. Das Ziel der Datenvorverarbeitung ist, die ausgewählten Daten für eine bessere Qualität aufzubereiten. Manche ausgewählten Daten haben wahrscheinlich unterschiedliche Formate, weil die Daten von verschiedenen Datenquellen gesammelt wurden [OD08, S. 12]. In diesem Abschnitt werden die gesamten Prozesse der Datenvorverarbeitung schrittweise genau erläutert.

2.3.1 Datenqualität

Wenn die Daten die Nutzungsvoraussetzungen erfüllen, wird das Thema Datenqualität berücksichtigt. Nach [HKP12, S. 84] wird die Datenqualität durch drei Hauptelemente definiert, nämlich die Genauigkeit, die Vollständigkeit und die Konsistenz. Im praktischen großen Datenbestand und im Dataset sind ungenaue, unvollständige und inkonsistente Daten üblich. Der Hauptgrund für Ungenauigkeiten ist wahrscheinlich die falsche Benutzung der Datenbeschaffungsinstrumente. Fehler an der Datenbereitstellung können entweder von Menschen oder von Computern gemacht werden. Die Fehler passieren manchmal auch bei der Datentransformation aufgrund technischer Beschränkungen, weil die Daten hier nicht mit hoher Genauigkeit transformiert werden können. Ungenaue Daten sind ein wichtiger Grund für Inkonsistenzen. Die Unvollständigkeit der Daten wird auch von praktischen Gründen verursacht. Manchmal sind die Attribute nicht immer verfügbar, für die die Menschen sich interessieren. Manche später wesentliche Daten werden am Anfang als unwichtige Daten erachtet und somit an der Datenbereitstellung herausgefiltert. Weiterhin können auch technische Geräteprobleme bestehen und manche relevanten Daten werden falsch gefiltert.

Nach [Net14] ist in der Realität ein „Konzept-Fehler“ ein wichtiger Grund der üblichen Datenqualitätsprobleme im kommerziellen DM-Prozess. Die konkreten Auswirkungen sind: Unterschiedliche Datenformate existieren in unterschiedlichen Datenquellen oder eine eventuell falsche Anwendung der Kennziffern. Diese Probleme machen die Zugriffe auf die Daten und das Datenverständnis schwieriger, z. B. Ein Attribut umfasst manchmal unterschiedliche Attributnamen in unterschiedlichen Datentabellen. Dieses Problem verursacht wahrscheinlich ein Missverständnis der Menschen und die Daten wurden gegebenenfalls falsch in der Datenbank eingegeben [Net14, S. 69].

2.3.2 Datenhomogenisierung

In diesem Abschnitt werden die Methoden der Datenbereinigung und der Datentransformation erläutert. Bei der Datenbereinigung gibt es auch zwei Untermethoden, nämlich die Bereinigung von fehlenden Werten und die Bereinigung von verrauschten Daten. Die Datentransformation konzentriert sich nur auf die Datentyptransformation für die Homogenisierung.

Datenbereinigung

In der Realität sind Daten tendenziell unvollständig, mit Fehlern oder Ausreißern behaftet und sogar inkonsistent. Die Datenbereinigung richtet ihren Schwerpunkt auf die Ausfüllung der fehlenden Werte und das Herausfinden der Rauschdaten, damit die Ausreißer und die inkonsistenten Daten identifiziert werden können. In diesem Abschnitt werden zwei Hauptaspekte für die Datenbereinigung behandelt, nämlich die Verfahren für die Bearbeitung der fehlenden Werte und der verrauschten Daten [HKP12, S. 88].

Fehlende Werte

Wenn man einen Datenanalyse-Auftrag angenommen hat, werden häufig folgende Methoden nach [HKP12, S. 88] angewendet, um die fehlenden Werte zu bereinigen.

1. Manuelle Ausfüllung der fehlenden Werte

Generell ist diese Methode zeitaufwendig und manchmal nicht einsetzbar, wenn das gegebene Dataset groß ist und innerhalb dessen zahlreichen Werte fehlen.

2. Ausfüllung der fehlenden Werte mit einer globalen Konstante

Dies ist die Ersetzung aller fehlenden Werte durch eine gleiche Konstante, wie beispielsweise ein Label ähnlich wie „Unbekannt". Aber es gibt auch systematische Nachteile, wenn man diese Methode in einer DM-Software einsetzt. Da alle ersetzten Werte gleich sind, werden die ersetzenden Werte durch das DM-Programm als eine interessante Kategorie falsch anerkannt. Deshalb ist diese Methode zwar einfach, aber nicht exakt.

3. Direkte Filterung der fehlenden Werten

Durch diese Methode werden die fehlenden Werte direkt gefiltert. Diese Methode ist aber nur geeignet für den Fall, dass die Gesamtsumme der fehlenden Werte nicht hoch ist. Sonst wird die Vollständigkeit des Datasets schwer beschädigt.

Es ist besonders wichtig aufzupassen, dass die fehlenden Werte in manchen Fällen nicht einen Fehler in den Daten implizieren! Ein Beispiel: Bei der Beantragung einer Kreditkarte wird der Bewerber aufgefordert, die Nummer seines Führerscheins anzugeben. Ein Kandidat ohne Führerschein wird dieses Feld frei lassen. Durch die Software werden diese Kandidaten als „darf sich nicht bewerben" erkannt. Idealerweise sollte jedes Attribut eine oder mehrere Regelungen bezüglich der *Null-Eingabe* haben. Dann wird die Frage, ob die *Null-Eingabe* erlaubt ist oder nicht, nach dieser Regelung spezifiziert. Die Felder können auch absichtlich leer bleiben, wenn sie in späteren Schritten eingegeben werden. Daher sollen das Design einer guten Datenbank und die gute Vorgehensweise des Datenbereitstellung am Anfang helfen, die Summe der fehlenden Werte oder Fehler zu minimieren [HKP12, S. 89].

Verrauschte Daten

Rauschen bedeutet einen zufälligen Fehler oder eine Varianz in einer gemessenen Variablen. Es entsteht hierbei die Frage: Wie können wir die Daten glätten, damit das Rauschen bereinigt werden kann? Im Folgenden werden einige Glättungsmethoden vorgestellt.

Klasseneinteilung (Binning)

Diese Methode glättet sortierte Datenwerte durch die Kommunikation mit ihren Nachbarn, nämlich die Werte in der Nähe des Datenwerts. Die sortierten Werte werden auf mehrere Körbe (Bins) verteilt. Diese Methode führt eine lokale Glättung durch, indem sie mit dem benachbarten Werten kommuniziert. Nun werden drei Binning-Verfahrens mithilfe der Tabelle 2.2 durch ein Beispiel von aufgeteilten Daten für die Preise (in Euro): 5, 9, 16, 22, 22, 25, 26, 29 und 35 vorgestellt. In dem ersten Verfahren werden die Daten nach dem Preis sortiert und in drei unterschiedliche Bins nach der gleichen Häufigkeit mit der Größe 3 aufgeteilt (das heißt jeder Bin enthält drei Werte). Beim zweiten Verfahren wird jeder Wert in einem Bin durch den Mittelwert von den drei Werten, die beim ersten Verfahren jedem Bin zugeordnet sind, ersetzt. Dieses Verfahren kann auch so funktionieren, dass jeder Wert eines Bins durch den Medianwert ersetzt wird. Bei dem dritten Verfahren werden die minimalen sowie maximalen Werte eines Bins als Grenzwert betrachtet. Generell ist es so, dass je breiter die Bins sind, desto größer der Glättungseffekt ist. Die Intervallbreite jedes Bins kann der Benutzer selbst mit einem speziellen Wert einstellen [HKP12, S. 90f.].

In dieser Masterarbeit werden die verrauschten Daten in dem späteren Modellierungsprozess mithilfe der Diskretisierungstechnik bereinigt, die auf der Binning-Methode basiert und sie wird im nächsten Abschnitt weiter erläutert.

Tabelle 2.2: Binning-Beispiel (nach [HKP12, S. 91])

Bin Nr. / Methode	Bin 1	Bin 2	Bin 3
Aufteilung zu Bins (gleiche Häufigkeit)	5, 9, 16	22, 22, 25	26, 29, 35
Glättung durch den Mittelwert der Bins	10, 10, 10	23, 23, 23	30, 30, 30
Glättung durch den Grenzwert der Bins	5, 5, 16	22, 22, 25	26, 26, 35

Neben der obengenannten Methode stehen auch andere Verfahren zur Bereinigung der verrauschten Daten zur Verfügung, z. B. Regression. Durch die Datenanalyse werden nur die obengenannten Verfahren nach dem Bedarf der Datenanalyse ausgewählt. Somit werden die anderen Verfahren in dieser Masterarbeit nicht mehr weiter erläutert.

Datenbereinigung als Prozess

Fehlende Werte, verrauschte Daten und inkonsistente Daten führen zu geringer Datengenauigkeit. Bis jetzt wurden zwei übliche Datenbereinigungsverfahren kurz erläutert. Aber Datenbereinigung ist wirklich eine komplexe Arbeit und soll als einen Prozess betrachtet und zu organisiert werden. Nach [HKP12, S. 91ff.] wird der Prozess der Datenbereinigung in folgenden drei Schritten durchgeführt.

Der erste Schritt des Datenbereinigungsprozesses ist die Entdeckung von Widersprüchlichkeiten, die normalerweise von einem mangelhaften Design der Datenbereitstellung, inkonsistenter Datenrepräsentation sowie inkonsistenter Benutzung der Kodes und durch Fehler in der Instrumentationsvorrichtung verursacht werden, die die Daten und Systemfehler erfassen. Der Anfangsschritt ist die Ausnutzung der Kenntnisse, die sich auf die Dateneigenschaften beziehen, z. B. die Metadaten. Die Metadaten beziehen sich auf die Frage, was die Datentypen und Datendomains von jedem Attribut sowie die akzeptierbaren Werte für jedes Attribut sind. Bei diesem Schritt wird ein eigenes Protokoll über die Daten manuell oder mithilfe von Werkzeugen erstellt. Mithilfe dieses Schritts können verrauschte Daten, Ausreißer und unormale Daten herausgefunden werden, die dann untersucht werden sollen.

Nach der Entdeckung von Widersprüchlichkeiten kommt die Datentransformation zum Einsatz. Die Hauptaufgabe dieses Prozesses ist die Definition und Applikation einer Reihe von Transformationen, um gefundene Widersprüchlichkeiten zu korrigieren. Dieser Prozess wird in einem späteren Kapitel genauer behandelt.

Die Entdeckung von Widersprüchlichkeiten und die Datentransformation sind zwei Prozesse, die sich immer wiederholen und fehleranfällig sowie zeitaufwendig sind. Manchmal werden mehr Widersprüchlichkeiten entdeckt nach der Datentransformation. Einige besondere Widersprüchlichkeiten werden hingegen erst nach der Transformation von anderen Widersprüchlichkeiten entdeckt. Aber die falschen neu erstellten Ausreißer können dann nur nach der kompletten Beendigung des Transformationsprozesses wieder geprüft werden.

Datentransformation

Manchmal enthält ein Dataset zwar keine extremen Ausreißer, aber es existieren doch potenzielle Ausreißer. Ein Ausreißer beeinflusst die Homogenität des Samples nicht stark, jedoch verursacht er große Abweichungen, die die zusammenfassende Kalkulation behindern kann [Pie15, S. 64]. Bei dem obengenannten Fall ist die originale Datenform nicht geeignet für den DM-Prozess. Mithilfe der Datentransformation wird die Datenform umgewandelt, damit der DM-Prozess mit der neuen Datenform durchgeführt werden kann [CL16, S. 216].

In der Realität umfasst jedes Attribut einen eigenen Wertebereich, der unterschiedlich zu anderen Attributen ist. Beispielsweise unterscheiden sich die Preise von verschiedenen Autoserien je nach Motorleistung. Der große Unterschied der Wertbereiche von den Attributen in den Daten verursacht eventuell eine Verfälschung des Ergebnisses der Datenanalyse. Der beobachtete Wertbereich und der gewünschte Wertbereich sind zwei Betrachtungspunkte, um eine passende Transformation auszuwählen [Run15, S. 35].
Häufig wird folgende Reihe von Transformationen nach [CL16] angepasst:

1. Datentypen
2. Konvertierungen oder Kodierung
3. Zeichenketten
4. Datumsangaben
5. Maßeinheiten und Skalierungen

Nun werden die Transformation des Datentyps vorgestellt, die speziell für die Datenhomogenisierung dienen.

Datentypen-Transformation für die Homogenisierung

Die Datentypentransformation ist ein wichtiger Bestandteil des DM-Prozesses, weil die Daten normalerweise nicht homogen sind und viele Typen enthalten. Zum Beispiel enthält ein demografisches Dataset nicht nur numerische Attribute, sondern auch gemischte Attribute. Die mehrfachen Datentypen verursachen eine verwirrende Situation für den Daten-Analysten, der jetzt mit einer schwierigen Herausforderung über das Design eines Algorithmus durch die ungeordneten Datentypen konfrontiert ist. Die gemischten Datentypen verhindern auch die Fähigkeit des Daten-Analysten, mit den vorhandenen Werkzeugen die Daten zu verarbeiten. Es soll beachtet werden, dass die Portierung der Datentypen einen Verlust der gegenständlichen Genauigkeit und Ausdrucksfähigkeit verursacht. Darunter werden drei Methoden für die Konvertierung zwischen verschiedenen Datentypen vorgestellt, die in den späteren Experimentprozess angewendet werden.

Numerische zu kategorischen Daten: Diskretisierung

Normalerweise beinhalten die Originaldaten verschiedene Datenformate, z. B. numerische Daten, nominale Daten, kontinuierliche Daten und diskrete Daten. Manchmal sind Daten zwar Nummern, aber sie haben keine numerische Bedeutung, z. B. die ID-Nummer von unterschiedlichen Produkten. In diesem Fall ist die Berechnung eines Durchschnittswerts oder einer Standardabweichung sinnlos. Deshalb ist der richtige Datentyp wichtig für den späteren DM-Prozess. Um den Datentyp von numerisch zu nominal zu transformieren, wird die Methode der Diskretisierung angewendet [GLH15, S. 245].

Diskretisierung ist eine wesentliche Vorverarbeitungstechnik und wird für Knowledge-Discovery- und DM-Aufgaben angewendet. Das Hauptziel ist, die kontinuierlichen Attribute zu diskreten Attributen zu transformieren durch die Kombination der kategorischen Werte zu Intervallen. Damit wird der Datentyp transformiert und die Daten werden homogenisiert [GLH15, S. 245]. Das Thema „Diskretisierung“ wird im nächsten Abschnitt weiter erläutert.

Kategorische zu numerischen Daten: Binarization
In manchen Fällen ist es erstrebenswert, numerische DM-Algorithmen auf die nominalen Daten anzuwenden. Da die binären Daten eine spezielle Form von numerischen und kategorischen Daten sind, ist es möglich, die nominalen Daten in binäre Form umzuwandeln. Somit können die numerischen Algorithmen auf die binären Daten angewendet werden. Wenn das nominale Attribut X unterschiedliche Attributwerte hat, werden dann X binäre entsprechende neue Attribute erstellt. Jedes neue binäre Attribut repräsentiert einen Attributwert der originalen nominalen Attribute. Die Attributwerte von den neuen Attributen sind „1“ und „0“. Der Attributwert wird mit dem Wert „1“ markiert, wenn dieses Attribut durch einen bestimmten Datensatz erfüllt wird und sonst wird der Wert „0“ übernommen [Agg15, S. 31]. Im späteren DM-Prozess wird diese Methode bei der „fachliche Kodierung“ im Abschnitt 4.1 angewendet.

Zeitserien zu numerischen Daten
Dieses besondere Transformationsverfahren ist nützlich, weil es die Anwendung von mehrdimensionalen Algorithmen auf die Zeitserien-Daten ermöglicht [Agg15, S. 32]. In dieser Masterarbeit werden die bestimmten Daten mit dem Datentyp „Date“ im Modellierungsprozess zum Datentyp „numerical“ transformiert, damit der Zeitabstand zwischen zwei Zeitserien-Attributen mithilfe des numerischen Algorithmus berechnet werden kann.

2.3.3 Datenaggregation

In diesem Abschnitt werden die Aggregationsmethoden vorgestellt, die im späteren Modellierungs- prozess angewendet werden. Zuerst wird die Definition der Aggregation kurz eingeführt. Danach werden die Datenintegrationsverfahren erläutert, die für die Erstellung des Zielformats notwendig sind. Anschließend werden die üblichen Stichproben-Methoden behandelt, die für die Datenkompression hilfreich sind. Danach wird die Datenanreicherung kurz vorgestellt und zum Schluss wird eine wichtige Datenaggregationsmethode, nämlich die Diskretisierung, mit Beispielen genau erläutert.

Definition von Aggregation
Manchmal gilt „weniger ist mehr“ und dieser Satz beschreibt genau den Fall der Aggregation, durch die zwei oder mehr Objekte zu einem einzigen Objekt kombiniert werden [TSK06, S. 45]. Im DM-Prozess kann das Wort „Objekt“ viele konkrete Bedeutungen haben, z. B. das Attribut, die Datenzeile, der Attributwert eines Attributes. Unter dem Begriff „Datenaggregation“ versteht man die Zusammenfassung von Datensätzen und Attributen von einer unteren Aggregationsstufe zu einer höheren Aggregationsstufe mithilfe der Aggregationsfunktion. Die Merkmale der höheren Aggregationsstufen werden ebenfalls aus einzelnen oder zahlreichen Merkmalen der unteren Aggregationsstufen mithilfe einer Funktion gebildet [Pet05, S. 60].
Im Folgenden werden einige Punkte nach [TSK06, S. 46] dargelegt und sie beziehen sich auf die Frage, warum eine Aggregation durchgeführt werden soll.

1. Ein kleiner Datensatz, der nach der Datenkompression resultiert, braucht weniger Speicherplatz und eine kürzere Durchlaufzeit. Somit ist es mit der Hilfe der Aggregation möglich, für die Daten die aufwendigen DM-Algorithmen anzuwenden.
2. Mithilfe der Aggregation können die Bereiche oder der Umfang von Daten geändert werden durch die Bereitstellung der Daten mit einer hohen statt einer niedrigen Blickbreite.
3. Nach der Aggregation ist das Verhalten von Gruppen der Objekte oder von Attributen häufig stabiler als von individuellen Objekten oder Attributen vor der Aggregation.

Umgekehrt bestehen bei der Aggregation auch Nachteile. Mithilfe der Datenaggregation wird die Datenmenge reduziert. Gleichzeitig ist die Datenaggregation immer von dem Problem „Datenverlust" begleitet. Die Einführung der zusätzlichen Merkmale wird als eine übliche Lösung betrachtet, die gegen das Prinzip der Datenkompression jedoch nicht verstößt [Pet05, S. 60].

Nach [Pet05] werden folgende Aggregationsstufen nach drei unterschiedlichen Aspekten als Beispiel aufgezählt:

- sachlich: Artikel > Kaufakt > Kunde > Kundengruppe > Gesamtmarkt
- räumlich: Kunde > Wohnblock > Ortsteil > Gemeinde > Vertriebsbezirk > Vertriebsregion
- zeitlich: Tag > Woche > Monat > Quartal > Jahr

Es ist besonders zu beachten, dass die Daten aus verschiedenen Aggregationsstufen manchmal verknüpft werden können. Bei diesen Fällen ist die Prüfung der Intraklassenvarianz sowie der Interklassenvarianz notwendig. Mithilfe der Datenaggregation wird die Streuung der Merkmale verringert, damit die Güte-Maße des DM-Modells mit der Einführung der neuen Aggregationen nach der systematischen Sicht verbessert werden [Pet05, S. 61].

Das nächste wichtige Thema bezieht sich auf die Frage, wie eine aggregierte Transaktion erstellt werden sollte. Die numerischen Attribute, wie beispielsweise der „Preis", werden normalerweise aggregiert durch Berechnung und Ersetzung des Summenwerts oder des durchschnittlichen Wertes. Die nominalen Attribute, wie beispielsweise „Produkt", können entweder ignoriert werden oder über beispielsweise die Produkte zusammengefasst werden, die am gleichen Ort verkauft werden [TSK06, S. 45].

Datenintegration

Ein schwieriges Problem von Data Mining ist die Beschaffung einzelnes Datasets aus den Informationen, die aus variierenden und verschiedenen Quellen stammen. Wenn der Integrationsprozess nicht richtig durchgeführt wird, werden Redundanzen und Widersprüchlichkeiten schnell aufkommen. Das Ergebnis ist, dass sich die Genauigkeit und die Geschwindigkeit der kommenden DM-Prozesse verringern. Die Anpassung der Schemata von unterschiedlichen Quellen verursacht jedoch ein bekanntes Problem, das in der Praxis häufig passiert: Widersprüchlichkeit und sich wiederholende Tupel sowie Redundanz und zusammenhängende Attribute sind Probleme, die auch später im Integrationsprozess des Datasets passieren können [GLH15, S. 40].

Ein wesentlicher Teil im Integrationsprozess ist es, ein Data Map zu erstellen. Das Data Map bezieht sich auf die Frage, wie jedes Dataset in einer allgemeinen Struktur organisiert werden kann, um ein Beispiel zu repräsentieren, das aus der Realität stammt. [GLH15, S. 40]. In dieser Masterarbeit wird ein ER-Modell zur Untersuchung von den Zusammenhängen zwischen unterschiedlichen Datentabellen im Abschnitt 3.1.3 aufgebaut.

Die Datenintegration enthält folgende Aspekte. Es kommt zuerst die Frage: Wie können die Objekte von verschiedenen Quellen zum Schema passen? Diese Frage ist die Kernaufgabe des Entitäten-Identifikationsproblems. Nach der Identifikation wird die Korrelation zwischen verschiedenen Attributen geprüft. Das genaue Verfahren ist die Durchführung eines Korrelationstests für Daten. Zum Schluss ist dann das Thema Tupel-Duplikation zu betrachten [HKP12, S. 94]. Im Folgenden werden die genauen Schritte der Datenintegration nach [HKP12, S. 94] ausführlich behandelt.

Entitäten-Identifikation

Während der Datenintegration sollen zahlreiche Aspekte überlegt werden, wobei Schema-Integration und Objektanpassung zwei wichtige Aspekte sind. Die Entitäten-Identifikation betrifft die Frage, wie Entitäten der realen Welt von mehrfachen Datenquellen äquivalent integriert werden können.

Die Metadaten jedes Attributs enthalten den Namen, den Mittelwert, die Datentypen und den erlaubten Wertbereich von Attributen. Solche Metadaten können eingesetzt werden zur Vermeidung von Fehlern in der Schema-Integration. Weiterhin können die Metadaten in der Datentransformation angewendet werden, um den Transformationsprozess zu unterstützen.

Bei der Datenstruktur muss besonders aufgepasst werden, wenn die Attribute von einer Datentabelle zu einer anderer während der Datenintegration integriert werden. Es soll gewährleistet werden, dass die funktionale Abhängigkeit und die referentielle Beschränkung von jedem Attribut der Quellensysteme mit denen von dem Zielsystem zusammenpassen.

Redundanz und Korrelationsanalyse

Die Redundanz ist ein anderes wichtiges Thema bei der Datenintegration. Ein Attribut wird als redundant betrachtet, wenn es von einem anderen Attribut oder von einer Reihe von Attributen abgeleitet werden kann. Eine Inkonsistenz in den Attributen oder der Dimensionsbenennung verursacht auch Datenredundanz. Manche Redundanzen können durch die Korrelationsanalyse entdeckt werden. Mithilfe dieser Analysemethode kann die Korrelationsintensität zwischen zwei gegebenen Attributen herausgefunden werden, die auf den verfügbaren Daten basiert. Für die nominalen Daten wird der Chi-Square-Test angewendet, während bei den numerischen Attributen die Korrelations-Koeffizient-Verfahren und Kovarianz-Verfahren eingesetzt werden [HKP12, S. 94].

Korrelations-Test für nominale Daten

Beim Fall von nominalen Daten kann die Korrelation zwischen zwei Attributen A und B mithilfe des Chi-Square-Tests festgelegt werden. Diese Methode wird später im Abschnitt „Feature Selection“ genauer behandelt.

Weil der Datentyp der Experimentdaten nominal ist, werden zwei anderen Verfahren “Korrelations-Koeffizient-Verfahren” und “Kovarianz-Verfahren” in dieser Masterarbeit nicht weiter behandelt.

Tupel-Duplikation

Neben dem Aufspüren von Redundanzen zwischen Attributen können Duplikationen auch auf der Tupel-Ebene aufgespürt werden. Die Inkonsistenz entsteht häufig zwischen verschiedenen Duplikaten wegen der ungenauen Datenbereitstellung oder des Hochladens von unvollständigen Datenwerten [HKP12, S. 98f.].

Daten-Stichprobe

Die Stichprobe wird von den Datensätzen der vorliegenden Datenbestände genommen, um ein viel kleineren Dataset zu erstellen. Der zentrale Vorteil der Stichprobe ist, dass sie einfach, intuitiv und relativ leicht zu implementieren ist. Die Auswahl der Stichprobentypen ändert sich je nach der vorliegenden Applikation [Agg15, S. 38]. Nachfolgend werden zwei Stichprobentypen vorgestellt.

Stichprobe für statische Daten

Es ist einfach, eine Stichprobe zu nehmen, wenn die kompletten Daten schon verfügbar sind, weil die Summe der Ausgangsdatenpunkte schon bekannt ist. Bei der unbefangenen Stichproben-Methode wird eine vorherbestimmte Bruchzahl f von den Datenpunkten ausgewählt und für die Analyse herangezogen. Die Implementierung dieses Prozesses kann nach zwei unterschiedlichen Verfahren durchgeführt werden, die sich auf die Anwendung des Schrittes „Ersetzung" beziehen [Agg15, S. 38].

Wenn die Stichprobe ohne den Schritt „Ersetzung" von einem Dataset mit N Datensätzen genommen wird, werden insgesamt $N*f$ Datensätze vom Dataset zufällig herausgenommen. In diesem Fall enthält die Stichprobe keine Duplikate, außer wenn das originale Dataset bereits die Duplikate enthält.

Wenn die Stichprobe mit dem Schritt „Ersetzung" von einem Dataset mit N Datensätzen genommen wird, werden die Datensätze sequentiell und unabhängig vom gesamten Dataset für insgesamt $N*f$ Male genommen. In diesem Fall ist die Erzeugung von Duplikaten möglich, weil durch das sequentielle Herausnahmenverfahren die gleichen Datensätze in die Stichprobe kommen können. Normalerweise wird die Stichprobe ohne den Schritt „Ersetzung" genommen, weil unnötige Duplikate die DM-Applikation behindern werden können [Agg15, S. 38]. Nachfolgend werden zwei Stichproben-Verfahren nach [Agg15, S. 38f.] kurz vorgestellt:

1. Befangene Stichprobe

Bei diesem Verfahren werden manche Daten wegen ihrer hohen Wichtigkeit für die Datenanalyse absichtlich hervorgehoben. Im späteren Experimentprozess werden die Experimentdaten wegen der hohen Datenmenge des Datenbestands mithilfe dieser Stichprobe-Methode genommen. Um die Eigenschaft des gesamten Datasets beizubehalten, werden die Stichproben mit dieser Methode durchgeführt. Im späteren Experimentprozess werden nach dem Bedarf des DM-Prozess einige Stichproben aus der integrierten Hauptdatentabelle genommen. Die Durchführung der befangenen Stichprobe erfolgt mittels der Software „SQL" und wird in Abschnitt 3.1.2 genau erläutert.

2. Geschichtete Stichprobe

In manchen Datasets können wichtige Bestandteile des gesamten Datasets wegen ihrer Seltenheit durch die Stichprobe nicht ausreichend repräsentiert werden. Deshalb ist der erste Schritt bei der geschichteten Stichprobe die Aufteilung der Daten zu einer Reihe von gewünschten Schichten. Dann werden die Stichproben von jeder Schicht basierend auf den vorbestimmten Proportionen auf eine applikationsspezifische Weise als unabhängig angenommen. In dieser Masterarbeit werden die Trainingsdaten für den späteren Experimentprozess mit 1000 Datenzeilen nach dieser Stichproben-Methode genommen, damit durch eine relative kleine Datenmenge relativ mehr Eigenschaften des gesamten Datasets widergespiegelt werden können.

Reservoir-Stichprobe für den Datenfluss (dynamisch)

Bei diesem Verfahren wird die Stichprobe mit k Punkten von einem Datenfluss dynamisch erhalten. Der Datenfluss ist ein extrem großes Volumen und deshalb kann die Reservoir-Stichprobe für den Datenfluss auf einer Festplatte, auf der dieser Datenfluss gespeichert wird, nicht durchgeführt werden. Weiterhin wächst die Datenmenge konstant, weil die dynamische Stichprobe immer neue Daten bekommt. Gleichzeitig werden manche Datenpunkte auch aus der Stichprobe verworfen. Somit arbeitet dieses Stichproben-Vorgehen zu jedem Zeitpunkt mit inkomplettem Wissen über die vorherige Historie des Datenflusses [Agg15, S. 39]. Weil die Experimentdaten statische Daten sind, wird dieses Verfahren nicht weiter behandelt.

Datenanreicherung

Im DM-Prozess ist es häufig notwendig, neue Attribute neben den originalen Attributen zu erstellen. Mithilfe der neu erstellten Attribute wird die Erfassungsfähigkeit der wichtigen Informationen in einem Dataset effizienter [TSK06, S. 55]. Die neuen Attribute können sowohl mithilfe der originalen Daten erstellt werden als auch von der externen Seite importiert werden. In diesem Abschnitt werden beide Fälle erläutert.

Zuerst werden zwei relevante Methoden zur Erstellung neuer Attribute mithilfe von originalen Daten vorgestellt, nämlich Attribut-Extraktion und Attribut-Konstruktion.

Attribut-Extraktion

Attribut-Extraktion bedeutet die Erstellung einer neuen Menge von Attributen aus den originalen Rohdaten, z. B. eine Menge von Fotos wird nach der Fragestellung klassifiziert, ob das Foto ein menschliches Gesicht enthält. Die Rohdaten sind jedoch ein Dataset von Pixeln des Fotos, die aber nicht geeignet für viele Typen von Klassifikationsalgorithmen sind. Wenn die Daten zur Bereitstellung der höheren Attribut-Stufe verarbeitet sind, können mehrere Klassifikationsverfahren auf die Daten nach der Verarbeitung angewendet werden, weil die höhere Stufe eine höhere Korrelation mit der Klassifikationsfragestellung haben soll. Zwar wird die Attribut-Extraktion in der Realität am häufigsten angewendet, aber gleichzeitig ist dieses Verfahren bereichsspeziell. Das heißt, dass die Attribut-Extraktionstechnik für einen speziellen Bereich schwer auf andere Bereiche angewendet werden kann [TSK06, S. 55].

Attribut-Konstruktion

Manchmal enthalten die Attribute innerhalb des originalen Datasets zwar die notwendigen Informationen für den DM-Prozess, aber die originale Datenform ist nicht geeignet für die DM-Algorithmen. In diesem Fall sind ein oder mehrere neue Attribute, die auf Grundlage der originalen Attribute konstruiert werden, nützlicher als die originalen Attribute [TSK06, S. 57]. Die genaue Funktionsweise ist, einige Mechanismen zu den originalen Algorithmen hinzufügen. Damit werden die originalen Attribute mit neuen Attributen zusammengesetzt. Das Hauptziel ist, die Genauigkeit des Attributes zu erhöhen und die Komplexität des Modells zu verringern [GLH15, S. 189].

Die Aufgabe der Attribut-Konstruktion im Bereich Datenvorverarbeitung ist die Anwendung von den Konstruktionsoperatoren auf die bereits vorhandenen Attributen, damit neue Attribute generiert werden, die auf die Beschreibung des Zielkonzepts besser angewendet werden können [GLH15, 189].

Jetzt wird die Funktionsweise der Anreicherung von neuen Attributen durch den Import von externer Seite mithilfe eines Beispiels vorgestellt.

Bei der Datenanalyse in den Themen Markt- und Absatzforschung besteht besonders das Problem, dass die existierende Datenbasis nach der fachlichen Sicht nicht ausreichend umfassend ist. Das heißt, dass Daten, die außerhalb des Unternehmens beschafft werden, in die Datenanalyse involviert werden müssen, z. B. bei Marktforschungsstudien. Die Daten von der externen Seite werden sich hauptsächlich aus regionalen oder demografischen Untersuchungen ergeben und werden durch Marktforscher global beschafft. Bei der Zusammenführung der existierenden Datenbasis und der unternehmensexternen Daten ist eine Prüfung der Zugehörigkeit von den Kundendatensätzen und den Kundensegmenten notwendig [Pet05, S. 57].

Datenanreicherung bedeutet nicht nur die Erstellung von neuen Attributen, sondern auch die Kompression der originalen Attribute. Dieser Aspekt ist ähnlich wie das Thema „Feature Selection“ und wird im Abschnitt 3.2 in dem Experimentprozess genau behandelt.

Diskretisierung

Diskretisierung ist eine der grundlegenden Datenkompressionstechniken. Durch den Diskretisierungsprozess werden die kontinuierlichen numerischen Attribute zu diskrete nominalen Attributen transformiert durch die Aggregation der originalen Attributwerte zu *a* unterschiedlichen diskreten Intervallen [GLH15, S. 245]. Nach der Diskretisierung werden die *a* diskretisierte Intervalle als die neuen Attributwerte des Attributes betrachtet. Die genaue Zahl von *a* wird durch die originalen Attribute und den Kontext festgelegt. Der Nachteil ist, dass die Datenschwankung innerhalb einer Teilmenge nach der Diskretisierung jedoch nicht mehr erkennbar wird. Deshalb verursacht die Datendiskretisierung einen Datenverlust, der aber für manche Applikationen nicht schlimm ist. Eine weitere große Herausforderung der Diskretisierung ist die uneinheitliche Verteilung der Daten in den unterschiedlichen Intervallen [Agg15, S. 30]. Um dieses Problem zu lösen, soll für jedes Intervall ein Sortierungsgewicht festgelegt werden und die Daten sollen nach unterschiedlichen Sortierungsgewichten in unterschiedlichen Intervallen diskretisiert werden, damit die Verteilungen von unterschiedlichen Intervallen ungefähr gleich sein können. In Abschnitt 4.1 wird ein ähnliches Problem bei der „fachlichen Kodierung“ auftreten und der Sortierungsprozess wird dort mit Experimentdaten und Experimentprozess genau behandelt.

Diskretisierungsprozess

Nach [GLH15, S. 249f.] wird der Diskretisierungsprozess typischerweise in vier Aspekte unterteilt:

1) Sortierung

Bei diesem Schritt werden die kontinuierlichen Werte eines Attributs entweder nach absteigender oder aufsteigender Reihenfolge sortiert. Die Sortierung muss nur einmal am Anfang von allen Diskretisierungsprozessen durchgeführt werden. Das heißt, die Sortierung ist ein zwingender Schritt der Diskretisierung.

2) Auswahl eines Schnittpunktes

Nach der Sortierung soll der beste Schnittpunkt innerhalb des Wertbereichs der Attribute gesucht werden, um den Wertbereich des Attributs aufzuteilen. Eine Evaluationsmethode oder

Funktion nach dem Klasse-Label wird angewendet zur Festlegung der Korrelation und zur Erlangung einer Leistungsverbesserung.

3) Aufteilung/Zusammenführung

Nach den Operationsmethoden der Diskretisierung können die Intervalle entweder aufgeteilt oder zusammengeführt werden. Für die Aufteilung werden alle realen Werte innerhalb des Wertbereichs eines Attributs als potenzielle Aufteilungspunkte betrachtet. Zuerst wird ein bester eingeschätzter Aufteilungspunkt gewählt und die Aufteilung eines kontinuierlichen Wertbereichs wird dann in zwei Partitionen unterteilt. Die gleichen Schritte werden wieder innerhalb der einzelnen Partitionen durchgeführt bis zur Erfüllung eines Stop-Kriteriums. Im Gegensatz dazu werden die besten Intervalle beim Fall der Zusammenführung ausgewählt, um die Zusammenführung bei jeder Iteration durchzuführen. Der Vorgang stoppt, wenn ein Stopp-Kriterium erfüllt wird. Mithilfe der Zusammenführung wird die Summe der Intervalle reduziert. Das Stop-Kriterium für beide Operationsmethoden soll nach dem Bedarf der Datenanalyse festgelegt werden.

4) Stop-Kriterium

In diesem Schritt werden die Stop-Kriterien festgelegt. Normalerweise werden zwei übliche Gedanken sorgfältig gegeneinander abgewogen: wenige Parameter einstellen für ein besseres Verständnis oder Konzentration nur auf die Genauigkeit oder die Konsistenz. Ein Stop-Kriterium kann entweder einfach, wie beispielsweise das Erreichen einer Zahl, oder komplex, beispielsweise durch eine Funktion, sein.

Repräsentative Diskretisierungsmethoden

Identisch zur Beschreibung der Diskretisierungsschritte gibt es hauptsächlich zwei repräsentative Diskretisierungsmethoden, nämlich die Aufteilungsmethode und die Zusammenführungsmethode.

Nach [Agg15, S. 30f.] werden zwei repräsentative Aufteilungsmethoden kurz erläutert:

1. *Gleiche Breite:* Diese Methode funktioniert nicht, wenn das Dataset über die unterschiedlichen Teilmengen nicht einheitlich verteilt ist. Um den aktuellen Wert der Teilmengen zu bestimmen, sollen minimale und maximale Werte für jedes Attribut festgelegt werden. Die Teilmenge [*min, max*] wird zu *X* Teilmengen mit jeweils gleicher Breite aufgeteilt.

2. Gleiche-Tiefe-Teilmenge: Bei diesem Fall werden die Teilmengen nach dem Kriterium ausgewählt, dass jede Teilmenge die gleiche Summe von Datensätzen hat. Das Ziel ist, jede Teilmenge mit einer gleichen Detailgenauigkeit anzubieten.

Für die Zusammenführungsmethode wird die Chi-Square-Statistik als eine repräsentative Methode vorgestellt. Die Chi-Square-Statistik ist eine statistische Methode und führt einen Signifikanz-Test über den Zusammenhang zwischen den Werten der normalen Attribute und den Werten der Label-Attribute durch. Die Grundprinzipien der Chi-Square-Statistik sind, dass die Häufigkeit des relativen Label-Attributs konsistent mit einem Attribut sein soll und zwei benachbarte Attribute voneinander unabhängig sein sollen. Durch diese Methode wird die Ähnlichkeit auf Basis des Signifikanz-Niveaus zwischen dem Label-Attribut und anderen Attributen festgelegt [GLH15, S. 263]. Die genaue Formel und die Berechnungsverfahren werden im folgenden Abschnitt „Feature Selection“ genauer erläutert.

2.3.4 Feature Selection

In diesem Abschnitt wird die letzte Datenvorverarbeitungsmethode „Feature Selection“ (FS) erläutert. Zuerst werden die Definition und die Typen von FS eingeführt. Dann werden die drei Hauptmethoden von FS kurz vorgestellt. Zum Schluss wird eine FS-Methode genau erläutert, die im späteren Modellierungsprozess angewendet wird.

Definition und Typen

Nicht alle Attribute sind gleich wichtig oder sinnvoll für die Prognose der verlangten Zielwerte. Einige Attribute hängen eng miteinander zusammen, wie beispielsweise das jährliche Gehalt und die Steuerausgaben [KD15, S. 26]. Eine hohe Anzahl von Dimensionen führt zu einer Reihe von Beschränkungen für die Fähigkeit der meisten DM-Algorithmen. Besonderer Grund dafür ist die Steigerung der Zahl der Rechnungsschritte wegen der hohen Zahl der Dimensionen. Eine effektive Lösung für das obengenannte Problem heißt Feature Selection, die heutzutage als eine der am häufigsten benutzten Techniken zur Behandlung mit den Themen „hohe Dimensionen“ und „Datenreduktion“ betrachtet wird [GLH15, S. 163].

Nach [GLH15, S. 163] wird der Begriff „Feature Selection“ wie folgt definiert:

“Feature Selection is a process that chooses an optimal subset of features according to a certain criterion”.

Das Kriterium entscheidet über die Details der Evaluation der Feature Subsets und muss je nach dem Ziel der Feature Selection ausgewählt werden. Zum Beispiel ist das optimale Feature Subset normalerweise ein minimales Subset, das die Prognose mit der besten Genauigkeit bieten kann [GLH15, S. 163].

Ziele von Feature Selection sind, die wichtigen Features im Dataset zu identifizieren und das Dataset um die anderen redundanten und irrelevanten Daten zu bereinigen. Mithilfe der FS wird die Anzahl der Dimensionen des Datasets reduziert, damit die DM-Algorithmen schnell durchgelaufen werden und ein besseres Ergebnis erhalten werden kann [GLH15, S. 164].

Die Durchführung der FS nach [GLH15, S. 164] bringt zahlreiche Vorteile, und zwar folgende:

1) Bereinigung von irrelevanten Daten
2) Steigerung der Prognosegenauigkeit des DM-Modells
3) Reduktion der Datenkosten
4) Erhöhung der DM-Effektivität, z. B. Reduktion der Reservenotwendigkeit und des Rechnungsaufwands.
5) Reduktion der Komplexität von der Beschreibung des Ergebnismodells und Erhöhung des Daten- und Modellverständnisses.

Feature Selection ist ein wichtiger Bestandteil des DM-Verfahrens, weil sie über die Qualität der Eingangsdaten entscheidet. Die Frage, welche Features relevant sind, hängt von der vorhandenen Applikation ab. Es gibt zwei unterschiedliche FS-Typen nach [Agg15, S. 40f.].

1. Unsupervised Feature Selection (UFS): Dieser Typ widmet sich der Aufgabe, das Rauschen und die Redundanzen der bestehenden Attribute zu bereinigen. UFS kann in verschiedenen Bereichen angewendet werden, aber nach ihrer Funktion passt die UFS am besten zur Clusteranalyse.

2. Supervised Feature Selection: Dieser Typ bezieht sich auf das Thema Datenklassifikation. Bei diesem Fall sind nur die Features am wichtigsten, die die Klasse prognostizieren können. Diese FS-Methode ist normalerweise geeignet für das Klassifikationsverfahren.

Feature Selection-Methode

Nach der Funktionsweise unterteilen FS-Methoden sich in zwei Verfahren, nämlich individuelle Evaluation und Subset-Evaluation. Die individuelle Evaluation wird auch Feature Ranking genannt. Mithilfe dieser Methode werden individuelle Features durch Verteilung ihres Gewichts und nach ihrem Grad der Zusammenhänge evaluiert, während die Subset-Evaluation und Feature-Subsets als Kandidaten auf Grundlage einer bestimmten Such-Strategie erstellt. Jeder Subset-Kandidat wird durch ein bestimmtes Evaluationsverfahren evaluiert und mit dem ehemaligen besten Kandidaten verglichen, auf den auch das gleiche Verfahren angewendet wurde. Das individuelle Evaluationsverfahren ist ungeeignet für die Bereinigung der redundanten Features, weil die redundanten Features normalerweise das gleiche Ranking haben. Im Gegensatz dazu können die redundanten Features mithilfe des Subset-Evaluations-Verfahrens durch die Kalkulation der Feature-Relevanz identifiziert werden [CMB15, S. 15].
Gemäß den Zusammenhängen zwischen den FS-Algorithmen und den induktiven Learning-Methoden wird FS nach [CMB15] hauptsächlich in drei Hauptmethoden unterteilt:

Filter

Diese Methode hängt von den generellen Charakteristiken der Daten ab. Bei dieser Methode wird der FS-Prozess als ein Vorverarbeitungsprozess betrachtet, der unabhängig von den induktiven Algorithmen ist. Die Vorteile dieser Methode sind der geringe Rechenaufwand und ihre gute Verallgemeinerungsfähigkeit. Im späteren Experiment wird diese Methode auf die Experimentdaten angewendet.

Wrappers

Diese Methode enthält einen Lernalgorithmus als eine sogenannte schwarze Box und ihre Prognosefähigkeit wird verwendet, um die relevanten Nutzungswerte von der Teilmenge der Variablen zu evaluieren. Bei dieser Methode wird die Lernmethode als ein Unterprogramm mit der Rechenbelastung für den FS-Algorithmus benutzt, um jede Teilmenge des Features zu evaluieren. Jedoch hilft die Interaktion mit dem Klassierer dabei, ein besseres Leistungsergebnis als das von den Filter-Methoden zu erhalten.

Embedded-Methode

Bei dieser Methode wird der FS-Prozess im Trainingsprozess durchgeführt und ist normalerweise speziell auf die Learning-Maschinen anzuwenden. Deshalb wird die Suche nach einer optimalen Teilmenge des Features innerhalb der Klassierer-Konstruktion gebildet und sie kann auch als Suche in der Kombination von Feature-Subsets und Hypothesen betrachtet werden. Es wird herausgefunden, dass mithilfe dieser Methode die Abhängigkeit einen geringeren Rechenaufwand als den von der Wrappers-Methode erfordern kann.

Angewendete Methode im späteren Experiment

Nun wird die im späteren Experimentprozess angewendete FS-Methode vorgestellt.

Chi-Square-Statistik

In vielen Fällen besteht das Dataset nur aus nominalen Attributen und eine gute Methode zur Unterscheidung zwischen den Attributen mit hoher Relevanz und den Attributen mit niedriger Relevanz ist die auf der Chi-Square-Statistik basierende Filtermethode. Mithilfe eines Chi-Square-Tests wird die Frage beantwortet, ob wirklich ein Zusammenhang zwischen beiden Attributen besteht. Wenn es viele Attribute gibt, kann die Chi-Square-Statistik auch für die Messung der Relevanz zwischen jedem Attribut und dem Label-Attribut angewendet werden [KD15, S. 360f.].

Jetzt wird die Funktionsweise der Chi-Square-Statistik genau erläutert. Bei der Chi-Square-Statistik werden die Ereignisse gezählt und die normalen Attribute werden mit dem Label-Attribut nach der Häufigkeit der Ereignisse verglichen. Mithilfe der Chi-Square-Statistik wird die Frage durch die Häufigkeit der Ereignisse geprüft, ob zwei Attribute in einer beliebigen Kombination korreliert sind. Durch die Prüfung der Korrelation wird die Eintrittswahrscheinlichkeit eines Ereignisses eines Attributs berechnet, wenn ein bestimmtes Ereignis eines anderen Attributs vorher festgelegt wird. Nach der Produktregel der Wahrscheinlichkeit wird die folgende Regel festgelegt: Wenn der Eintritt eines Ereignisses *A* unabhängig vom Ereignis *B* erfolgt, beträgt der Wahrscheinlichkeitswert (p_A*p_B), wenn *A* und *B* gleichzeitig passieren. Der nächste Schritt ist die Umrechnung der obengenannten Wahrscheinlichkeit zu einer zu erwartenden Häufigkeit nach diesem Produkt (p_A*p_B*N), wobei *N* die Summe der Ereignisse im Dataset ist [KD15, S. 361].

Nach der Berechnung der einzelnen Eintrittsmöglichkeiten wird eine Tabelle über die beobachteten Häufigkeiten erstellt, die „Kontingenztabelle“ genannt wird. Die letzte Spalte und Zeile dienen jeweils für die Überprüfung der Summe von den entsprechenden Spalten und Zeilen. Mithilfe der Kontingenztabelle wird weiterhin eine entsprechende Häufigkeitstabelle durch die Formel der zu erwartenden Häufigkeit (p_A*p_B*N) erstellt. Mithilfe der beiden Tabellen können die Unterschiede zwischen der beobachteten Häufigkeit und der erwarteten Häufigkeit jedes Attributs verglichen werden. Die Formel der Chi-Square-Statistik steht für das Aufsummieren der Differenzen aller Zellen zwischen beobachteten und erwarteten Häufigkeiten. Nachstehend wird die Formel der Chi-Square-Statistik nach [HKP12, S. 95] genauer vorgestellt:

Angenommen, *A* hat *c* unterschiedliche Werte, mit den Namen*:* $a_1, a_2, \ldots, a_c$. Der Wert *B* hat *r* unterschiedliche Werte mit den Namen*:* $b_1, b_2, \ldots, b_r$. Das Daten-Tupel, das durch die Werte *A* und *B* beschrieben wird, kann als eine Zufälligkeitstabelle aufgezeichnet werden. Die Spalten der Tabelle werden auf Basis von „*c*“ Werten von *A* und die Zeilen auf Basis von „*r*“ Werten von *B* aufgebaut. Nun wird (A_i,B_j) als das „Joint Event“ angezeigt. Die Werte des Attributs *A* werden vom Wert „a_i“ übernommen und die Werte der Attribute *B* werden vom Wert „b_j“ übernommen, d. h. $A = a_i$, $B = b_j$. Jedes mögliche (A_i,B_j) „Joint Event“ hat seine eigene Position in der Tabelle. Der Wert von X^2 kann mithilfe der Formel 2.1 berechnet werden.

$$X^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij}-e_{ij})^2}{e_{ij}} \qquad \textbf{(Formel 2.1)}$$

wobei „o_{ij}“ die beobachtete Häufigkeit von Joint Events ist und „e_{ij}“ die erwartete Häufigkeit von (A_i, B_j) bedeutet. Der Wert von „e_{ij}“ kann mithilfe der Formel 2.2 berechnet werden:

$$e_{ij} = \frac{count(A=a_i) \times count(B=b_j)}{n} \quad \textbf{(Formel 2.2)}$$

wobei der Wert „*n*“ die Summe des Daten-Tupels ist. *„count(A=a_i)“* ist die Summe des Tupels, die der Wert a_i für Attribut *A* hat, während *„count (B=b_j)“* die Summe des Tupels ist, die der Wert *„b_j“* für Attribut *B* hat.

Diese Methode prüft die Hypothese, dass der Wert *A* und der Wert *B* voneinander unabhängig sind. Das heißt, dass es keine Korrelation zwischen Wert *A* und Wert *B* gibt [HKP12, S. 95].

Um die genaue Funktionsweise der „Chi-Square-Statistik“-Methode besser zu erklären, wird ein einfaches Beispiel mit genauen Rechnungsschritten angezeigt.

Die Hypothese dieses Beispiels ist, dass das Thema „Vorliebe für Bier“ unabhängig vom Geschlecht ist. Eine Umfrage unter 2000 Menschen wird durchgeführt und das Geschlecht aller Teilnehmer wird vermerkt. Jeder wird befragt, ob Bier seine oder ihre Vorliebe ist. Deshalb hat dieses Beispiel zwei Attribute, nämlich das Geschlecht und die Vorliebe von Bier. Durch die Umfrage wird die Häufigkeit jedes möglichen „Join Event“ mithilfe der Tabelle 2.3 zusammengefasst.

Tabelle 2.3: Kontingenztabelle für das Beispiel der Chi-Square-Statistik (nach [BS13, S. 62])

	Männer	Frau	Gesamt
Bier ist Vorliebe.	1000	400	1400
Bier ist keine Vorliebe.	200	400	600
Gesamt	1200	800	2000

Mithilfe der oben gezeigten Formel 2.2 werden die Werte der Erwartungshäufigkeit für alle Datenfelder in der Tabelle 2.3 berechnet, zum Beispiel die Erwartungshäufigkeit für das Datenfeld *(Männer, Bier ist Vorliebe)*:

$$e_{11} = \frac{count(Männer) \times count(Bier\ ist\ Vorliebe)}{n} = \frac{1200 \times 1400}{2000} = 840$$

Nach diesem Verfahren werden die anderen drei Werte der Erwartungshäufigkeit berechnet: $e_{12} = 560$, $e_{21} = 360$ und $e_{22} = 240$. Es soll besonders aufgepasst werden, dass die Gesamtsumme der Erwartungshäufigkeit von allen Datenfeldern einer Datenzeile identisch zur Gesamtsumme der beobachteten Häufigkeit dieser Datenzeile sein muss. Dieses Prinzip ist auch geeignet für die Spalten. Nach der Formel 2.1 wird der X^2-Wert berechnet:

$$X^2 = \frac{(1000-840)^2}{840} + \frac{(400-560)^2}{560} + \frac{(200-360)^2}{360} + \frac{(400-240)^2}{240} \approx 253{,}97$$

Der Freiheitsgrad dieser Tabelle ist (2-1) × (2-1) = 1. Wenn der Freiheitsgrad gleich 1 ist, ist der notwendige X^2-Wert zur Ablehnung der Hypothese auf dem 0,001-Signifikanz-Niveau 10,828. Weil der berechnete Wert höher als 10,828 ist, wird die anfängliche Hypothese abgelehnt. Es wird zusammengefasst, dass das Thema „Vorliebe für Bier“ abhängig vom Geschlecht ist.

2.4 Clusteranalyse

Nach der Datenvorverarbeitung sind die Daten bereinigt und transformiert und stehen bereit, um mit ihnen den DM-Prozess durchzuführen. In diesem Abschnitt wird nach der Fragestellung ein wichtiges DM-Verfahren erläutert, nämlich die Clusteranalyse. Zuerst werden die Grundbegriffe der Clusteranalyse eingeführt. Danach wird die Clusteranalyse im Kontext des DM-Prozesses unter unterschiedlichen Fragestellungen weiter erläutert. Dann werden drei grundlegende Clusteranalyse-Methoden kurz erläutert und die dazu gehörigen zwei Algorithmen genau erklärt, die im späteren Experiment angewendet werden. Zusätzlich werden die eventuellen Modifikationen von beiden Algorithmen bei der Anwendung auf die nominalen Daten angezeigt. Zum Schluss werden die Clustervalidierungsmethoden vorgestellt.

2.4.1 Grundbegriffe

Die *Clusteranalyse* oder einfach *Clustering* ist der Prozess der Partitionierung einer Menge von Datenobjekten zu einigen Untermengen. Jede Datenmenge ist ein *Cluster*. Die Daten innerhalb eines Clusters sind ähnlich zueinander, während die Daten unähnlich zu den Datenobjekten sind, die zur anderen Clustern gehören. Die Cluster werden durch die Clusteranalyse herausgefunden, wobei unterschiedliche Clusteranalyse-Methoden verschiedene Clusteranalysen mit gleichem Dataset durchführen können. Die Partitionierung der Datenmenge wird nicht manuell, sondern durch Clusteranalyse-Algorithmen durchgeführt. Deshalb ist die Clusteranalyse nützlich in dem Fall, dass zuvor unbekannte Gruppen innerhalb der Daten herausgefunden werden [HKP12, S. 444].

Die Clusteranalyse ist besonders geeignet für die Gruppierung der unstrukturierten nominal und metrisch skalierten Daten [Küp99, S. 70]. Mithilfe der Clusteranalyse werden die Daten in Gruppen (Cluster) aufgeteilt, die bedeutungsvoll, nützlich oder beides sind. Wenn das Ziel die Aufteilung der Daten in bedeutungsvolle Gruppen ist, dann sollen die Cluster die Naturstruktur der Daten erfassen. Bei manchen Fällen ist die Clusteranalyse jedoch nur ein sinvoller Anfangspunkt für die anderen DM-Verfahren, z.B. Klassifikationsverfahren. Entweder für das Verständnis oder für die Nützlichkeit hat die Clusteranalyse in verschiedenen Bereichen seit langer Zeit eine wichtige Rolle gespielt, z.B. die Psychologie und andere soziale Wissenschaften, Biologie, Statistik, Pattern-Erkennung, Informationsabruf, Maschinenlernen und auch Data Mining [TSK06, S. 487]. Nach [TSK06] werden zwei Hauptanwendungen der Clusteranalyse in der Praxis vorgestellt.

Clusteranalyse für das Datenverständnis
Cluster oder konzeptionell bedeutungsvolle Gruppen von Objekten, die gemeinsame Merkmale haben, spielen eine wichtige Rolle bei der Analyse und Beschreibung der Dinge auf der Welt. Tatsächlich sind die Menschen befähigt, Objekte in Gruppen (Cluster) aufzuteilen und besondere Objekte den Gruppen zuzuordnen. Wenn die Clusteranalyse dem Datenverständnis dient, sind die Cluster potenzielle Klassen und die Clusteranalyse ist eine technische Studie für das automatische Herausfinden der Klassen. Die Anwendungsbereiche in diesem Fall sind beispielsweise die Biologie, der Informationsabruf, das Klima, die Psychologie und die Medizin sowie der Businessbereich.

Clusteranalyse für den Prototyp
In diesem Fall bietet die Clusteranalyse eine Abstraktion von individuellen Datenobjekten zu dem Cluster, zu dem die Datenobjekte gehören. Zusätzlich kennzeichnen einige Clustertechniken jeden Cluster hinsichtlich des Prototyps, der der Vertreter von anderen Objekten im Cluster ist. Solche Cluster-Prototypen können als Basis auf mehrere Datenanalyse- oder Datenverarbeitungstechniken angewendet werden. Wenn die Clusteranalyse der Nützlichkeit dient, ist sie eine technische Studie für die Identifizierung von den Cluster-Prototypen, die am ehesten repräsentativ sind. Die Anwendungsbereiche in diesem Fall sind beispielsweise die Datensummierung, die Datenkompression und die Suche nach den nächsten Nachbarn.

2.4.2 Clusteranalyse im Data Mining

Als ein DM-Verfahren kann die Clusteranalyse als ein separates Werkzeug angewendet werden, um einen Einblick in die Datenverteilung zu erhalten, die Merkmale jedes Clusters zu beobachten und eine besondere Menge von Clustern für die spätere Analyse zu fokussieren. Zudem kann die Clusteranalyse als ein Vorverarbeitungsschritt auch die anderen Algorithmen bedienen, wie beispielsweise die Feature Subset Selection und Klassifikation, damit die Algorithmen direkt mithilfe der entdeckten Cluster durchführen können [HKP12, S. 445].

In manchen Applikationen wird die Clusteranalyse auch als *Datensegmentation* bezeichnet, weil ein großes Dataset mithilfe der Clusteranalyse nach der *Gemeinsamkeit der Daten* in verschiedene Gruppen aufgeteilt wird. Die Clusteranalyse kann auch für die Entdeckung von Ausreißern angewendet werden, die interessanter als die allgemeinen Daten sind [HKP12, S. 445].

Im Maschinenlernen wird das Klassifikationsverfahren als *supervised learning* betrachtet, weil die Klassen-Labelinformationen vorher gegeben werden. Das heißt, der Lernalgorithmus wird überwacht. Im Gegensatz dazu wird das Clustering als *unsupervised learning* bezeichnet, weil die Klassen-Labelinformationen vorher nicht gegeben sind.

Aus diesem Grund ist die Form der Clusteranalyse eher ein Lernen durch Beobachtung als ein Lernen durch Beispiele [HKP12, S. 445].

Vergleich der Clusteranalyse mit dem Klassifikationsverfahren
Ein Cluster ist eine Gruppe von Datenobjekten, die einander gleich innerhalb eines Clusters und ungleich mit den Datenobjekten außerhalb des Clusters sind. Deshalb wird ein Cluster von den Datenobjekten auch als eine implizite Klasse betrachtet. In diesem Sinne wird die Clusteranalyse manchmal auch automatische Klassifikation genannt [HKP12, S. 445]. Zwar braucht die Clusteranalyse keine bestimmte Lernrichtung (*unsupervised*-Lernverfahren), aber sie teilt die methodologische Grundlage mit dem Klassifikationsverfahren. Das heißt, dass die meisten mathematischen Modelle von Klassifikationsverfahren auf die Clusteranalyse angewendet werden können [SZT+15, S. 4].

Im Vergleich mit dem Klassifikationsverfahren ist das Klassen-Label (oder Gruppen-ID) aller Daten bei der Clusteranalyse unbekannt. Die Gruppierung soll selbst entdeckt werden. Wenn große Mengen von Daten gegeben sind und viele Attribute die Datenprofile beschreiben, ist es aufwendig oder sogar nicht möglich, die Daten manuell zu analysieren, die Kenntnisse manuell zu extrahieren und eine Methode für die Aufteilung der Daten in verschiedenen strategischen Gruppen manuell zu entwickeln. In diesem Fall wird das Werkzeug Clusteranalyse eingesetzt [HKP12, S. 443].

Nach [MR15, S. 4] wird das Clusteranalyse-Verfahren normalerweise in folgenden vier Situationen statt des *supervised*-Klassifikationsverfahrens in der Datenanalyse angewendet:

1) wenn die Einsetzung des Labels im Dataset aufwendig oder unmöglich ist,
2) wenn die verfügbaren Label von den Daten missverständlich sind,
3) wenn das Verständnis der Dateneigenschaften verbessert werden muss,
4) wenn die Summe der Daten reduziert und die originalen Daten transformiert werden sollen.

Aus den obengenannten Gründen wird die Clusteranalyse als ein grundlegendes Werkzeug für DM, Dokumentenabruf, Image-Segmentation und Pattern-Klassifikation betrachtet.

Anwendung der Clusteranalyse für die Datenvorverarbeitung

Das Clusteranalyse-Verfahren kann als eine Datenkompressionstechnik angewendet werden. Die Ergebnisse der Clusteranalyse sind die unterschiedlichen Cluster für jedes Dataset und sie können als die Datenbereitstellung auf die anderen prognostischen DM-Verfahrens angewendet werden. Deshalb kann das Clusteranalyse-Verfahren als eine Vorverarbeitungstechnik für die anderen DM-Prozesse verwendet werden. Generell werden Clusteranalyse-Verfahren nach [KD15, S. 218f.] in nachfolgend genannten zwei Bereichen der Datenvorverarbeitung eingesetzt.

1. Clusteranalyse für die Reduktion der Dimensionen

In einem *n*-dimensionalen Dataset ist die Rechnungskomplexität proportional zur Summe der Dimensionen. Mithilfe der Clusteranalyse können die *n*-dimensionalen Attribute konvertiert oder reduziert zu einem kategorischen Attribut werden: *„Cluster-ID“*. Die Konvertierung reduziert zwar die Komplexität, aber gleichzeitig verursacht sie Datenverlust wegen der Reduktion der Dimensionen zu einem einzigen Attribut.

2. Clusteranalyse für die Reduktion der Datenobjekte (Datensätze)

Angenommen, dass die Kundenzahl eines Unternehmens in die Millionen geht und die Summe von Clustern auf „100“ festgelegt wird. Für jede dieser 100 Cluster-Gruppen wird ein Prototyp-Kunde identifiziert, der die Merkmale von allen Kunden in dieser Clustergruppe vertreten kann. Dieser Prototyp-Kunde kann entweder ein tatsächlicher Kunde oder ein fiktionaler Kunde mit typischen Merkmalen der Kunden in der Clustergruppe sein. Der Prototyp eines Clusters ist der allgemeinste Vertreter von allen Datenobjekten und er kann auch ein neues Objekt sein, wobei sein Attributwert der durchschnittliche Wert von allen Objekten im Cluster ist. Für die kategorischen Attribute soll der Attributwert des fiktionalen Prototyps mit dem Wert eingesetzt werden, der am häufigsten im Cluster erscheint. Die Reduktion der Kundendaten von Millionen zu 100 Prototyp-Daten bringt einen eindeutigen Vorteil: Statt der Verarbeitung von Millionen Datensätze müssen nur 100 Prototyp-Werte für die spätere Klassifikationsaufgabe verarbeitet werden. Mithilfe dieser Methode wird die Summe der Datenobjekte des Datasets stark reduziert.

Auswahl der Clusteranalyse-Algorithmen

Nach [TSK06] sind es folgende typische Faktoren, die bei der Auswahl eines Clusteranalyse-Algorithmus berücksichtigt werden sollen.

Typ der Clusteranalyse

Der Clusteranalyse-Typ muss zu dem praktischen Anwendungsbereich passen und die Algorithmen von unterschiedlichen Clusteranalyse-Methoden sind ein wichtiges Beurteilungskriteri-

um. Für ein Klassifizierungsproblem ist die Anwendung der hierarchischen Methoden geeignet. Wenn sich die Clusteranalyse-Aufgabe auf die Summe bezieht, ist die Partitions-Methode eine typische Lösung.

Typ des Clusters
Identisch zum ersten Punkt muss der Clustertyp in den praktischen Anwendungsbereich passen. Es gibt hauptsächlich drei Clustertypen, nämlich prototyp-, graphen- und dichtebasierende Clustertypen. Die ersten zwei Clustertypen produzieren normalerweise kugelförmige Cluster und in diesem Fall befindet sich jedes Objekt in der Nähe des Prototyps. Die beiden Clustertypen werden angewendet zur Summierung der Daten und zur Reduktion der Dataset-Größe. Im Gegensatz dazu produziert der dichtebasierende Clustertyp keine kugelförmigen Cluster und in diesem Fall gibt es viele Objekte, die nicht ähnlich wie die anderen Objekte sind.

Summe der Attribute
Die Clusteranalyse-Algorithmen für niedrige oder moderate Summen der Dimensionen funktionieren unter Umständen nicht gut für Datasets mit hohen Dimensionen. Wenn die Clusteranalyse-Algorithmen nicht richtig angewendet werden, können die Ergebnisse die richtige Datenstruktur unter Umständen nicht richtig repräsentieren.

Datenaufbereitung für die Clusteranalyse
Der Aufbereitungsprozess für die Clusteranalyse hängt vom Konzept der Distanz oder Ähnlichkeitsmaße ab, wobei Skalierung und Gewichtung eine besonders wichtige Rolle spielen. Mithilfe der Skalierung werden die Werte der Variablen an den Zustand der Realität angepasst, so dass die unterschiedlichen Variablen mit verschiedenen Einheiten oder innerhalb unterschiedlicher Wertbereiche gemessen werden. Mithilfe der Gewichtung werden unterschiedliche Variablen mit verschiedenen Gewichtungen angepasst, weil manche Variablen wichtiger als andere sind [LB11, S. 495].

2.4.3 Ähnlichkeitsmaße

Weil der Datentyp der Firmendaten nominal ist, sind die Abstandsmaß-Methoden bei diesem Fall nicht geeignet für die Clusteranalyse. Deshalb soll die Ähnlichkeitsmaß-Methode zur Berechnung der Ähnlichkeit zwischen unterschiedlichen Datenpunkten eingesetzt werden. In diesem Abschnitt wird das Thema „Ähnlichkeitsmaße" behandelt. Zuerst wird ihre Definition erläutert. Anschließend wird eine Ähnlichkeitsmaß-Methode für die binären Daten beschrieben. Zum Schluss wird eine Ähnlichkeitsmaß-Methode für die nominalen Daten vorgestellt.

Definition

Nach [BS13, S. 60] wird der Begriff „*Ähnlichkeit*" wie folgt definiert: „*Ähnlichkeit*" quantifiziert hauptsächlich die Beziehung zwischen unterschiedlichen Attributen von unterschiedlichen Datenobjekten. Angenommen, dass es zwei Objekte *i* und *j* gibt. Hierbei soll die Ähnlichkeit zwischen diesen zwei Objekten mit dem Zeichen s_{ij} bezeichnet werden. Der Wert vom s_{ij} hängt hauptsächlich von der Messungsskalierung und dem Datentyp ab. Andererseits messen der *Abstand* oder die *Unähnlichkeit* den Unterschied zwischen zwei Punkten auf Grundlage ihrer Attributwerte. Angenommen, dass der normalisierte Abstand und die Ähnlichkeit zwi-

schen zwei Objekten *i* und *j* jeweils d_{ij} und s_{ij} sind, so wird der Zusammenhang zwischen beiden Werten mithilfe der Formel 2.3 beschrieben:

$$s_{ij} = 1 - d_{ij} \qquad \textbf{(Formel 2.3)}$$

Die Clusteranalyse fängt mit der Auswahl eines Ähnlichkeitsmaßes und eine Menge von Variablen an. Es wird damit entschieden, mit welcher Ähnlichkeitsmaß-Methode in den Clusteralgorithmen berechnet werden sollen. Es soll bei der Auswahl der Ähnlichkeitsmaß-Methode aufgepasst werden, weil unterschiedliche Maße oft zu unterschiedlichen Clusteranalyseergebnissen führen können [BKN08, S. 404].

Ähnlichkeitsmaße für binäre Daten

Die binären Daten umfassen nur zwei unterschiedliche Datenwerte, z. B. 0 oder 1, yes oder no, true oder false. Jeder binäre Datenpunkt repräsentiert einen Vektor von den binären Variablen. Um die Ähnlichkeit von zwei binären Datenpunkten zu messen, soll zuerst die gesamte Anzahl von möglichen Ereignissen jedes Datenwertes gezählt werden [BS13, S. 61].

Nach [CPS+07, S. 258] wird der Vektor eines binären Datenpunkts in Formel 2.4 definiert: Angenommen, dass *x* und *y* zwei binäre Vektoren sind:

$$x = [x_1, x_2 \ldots . x_n]^T \qquad y = [y_1, y_2 \ldots . y_n]^T \qquad \textbf{(Formel 2.4)}$$

Nach [BS13, S. 61] werden die vier möglichen Ereignisse von binären Daten aufgezählt.

m_{00} = Gesamtsumme von Attributen, die den Wert 0 in beiden Objekten haben,

m_{01} = Gesamtsumme von Attributen, die den Wert 0 für das *i*-te Objekt und den Wert 1 für das *j*-te Objekt haben,

m_{10} = Gesamtsumme von Attributen, die den Wert 1 für das *i*-te Objekt und den Wert 0 für das *j*-te Objekt haben,

m_{11} = Gesamtsumme von Attributen, die den Wert 1 in beiden Objekten haben,

Die Gesamtsumme der Attribute beträgt $F = m_{00} + m_{01} + m_{10} + m_{11}$. **(Formel 2.5)**

Zum Beispiel: Für zwei binäre Punkte $p_1 = (1,0,1)$ und $p_2 = (0,0,1)$ betragen $m_{00} = 1$, $m_{01} = 0$, $m_{10} = 1$ und $m_{11} = 1$

In der Tabelle 2.4 wird die Konfusionsmatrix von beiden Punkten angezeigt:

Tabelle 2.4: Konfusionsmatrix der Ähnlichkeitsmaße für binäre Daten (nach [BS13, S. 62])

p_1	p_2	
	0	1
0	m_{00}	m_{01}
1	m_{10}	m_{11}

Jaccard-Koeffizient

In späteren Experimentprozess dieser Masterarbeit wird der „Jaccard-Koeffizient“ als die Ähnlichkeitsmaße auf die binären Daten zur Clusteranalyse nach der „fachlichen Kodierung“ angewendet. Hier wird dieses Ähnlichkeitsmaße nach [BS13, S. 62] kurz vorgestellt.

Diese Methode ist geeignet für die binären Daten, deren Datenwerte „0“ und „1“ ungleiche Häufigkeit bei den unterschiedlichen oben gezeigten Ereignissen haben. Für dieses Ähnlichkeitsmaße gilt die Formel 2.6:

$$s_{ij} = \frac{m_{11}}{m_{11}+m_{01}+m_{10}} \qquad \textbf{(Formel 2.6)}$$

In diesem Ähnlichkeitsmaße wird das Ereignis m_{00} nicht berücksichtigt. Nach der „fachlichen Kodierung“ des späteren Experimentprozesses erscheint ein Attribut bei einem Datensatz nur dann, wenn das Datenfeld mit dem Wert „1“ markiert ist. Das Ereignis „m_{00}“ ist nicht relevant für die Berechnung der gesamten Ähnlichkeiten. Deshalb ist dieses Ähnlichkeitsmaße geeignet für den späteren DM-Prozess. Ein Gegenbeispiel sind die binären Daten zum „Geschlecht“. In diesem Fall sind beide Datenwerte „Mann“ und „Frau“ relevant für die Berechnung der Ähnlichkeit und Jaccard-Koeffizient ist somit nicht geeignet.

Ähnlichkeitsmaße für nominale Daten

Eine bekannte Methode zur Messung der Unähnlichkeit von nominalen Daten ist die „Simple Matching Distance“. Unten werden die Formeln dieser Methode nach [KZ15, S. 1114] gezeigt.

Angenommen, dass *x* und *y* zwei nominale Datenpunkte sind. Die „Simple Matching Distance“ zwischen *x* und *y* wird in Formel 2.7 definiert:

$$\delta(x,y) = \begin{cases} 0, falls\ x = y \\ 1, falls\ x \neq y \end{cases} \qquad \textbf{(Formel 2.7)}$$

Für die beiden nominalen Datenpunkte *x* und *y* mit *m* Attributen wird die Unähnlichkeit zwischen den beiden Datenpunkten mithilfe der Formel 2.8 berechnet:

$$d_{sim}(x,y) = \sum_{j=1}^{m} \delta(x_j, y_j) \qquad \textbf{(Formel 2.8)}$$

Mit dem Ergebnis der Unähnlichkeit kann der Wert der Ähnlichkeit nach der Formel 2.3 berechnet werden.

2.4.4 Grundlegende Clusteranalyse-Methoden

Generell können die wesentlichen fundamentalen Clusteralgorithmen nach [HKP12, S. 448f.] in folgende Methoden klassifiziert werden.

Partitions-Methode

Wenn ein Dataset von *n* Objekten gegeben ist, werden *k* Partitionen von den Daten durch die Partitions-Methode aufgebaut. Jede Partition vertritt ein Cluster und $k \leq n$. Das heißt: Mithilfe

dieser Methode werden die kompletten Daten in *k* Gruppen aufgeteilt und jede Gruppe muss mindestens ein Objekt enthalten.

Anders gesagt, wird durch diese Methode eine einstufige Partitionierung im Dataset aufgebaut. Die grundlegende Partitions-Methode adoptiert typischerweise die individuelle Cluster-Separation. Das heißt, dass jedes Objekt genau zu einer Gruppe gehören muss.

Die meisten Partitions-Methoden stehen auf Basis der Distanz. Die Summe der Partitionen, die aufgebaut werden sollen, wird am Anfang gegeben, dann wird eine anfängliche Partitionierung durch die Partitions-Methode erstellt. Als nächster Schritt wird die iterative Verlagerungstechnik angewendet und es wird versucht, die Partitionierung durch die Bewegung der Objekte von einer Gruppe zu einer anderen Gruppe zu verbessern. Das allgemeine Kriterium einer guten Partition ist, dass die Objekte mit einem gleichen Cluster nah beieinanderliegen und die Objekte mit ungleichen Clustern weit auseinander zu stehen kommen. Es bestehen verschiedene Kriterien zur Beurteilung der Qualität der Partition.

Für die Partitions-Methode ist die Erzielung der globalen Optimalität wegen des hohen Rechenaufwandes nicht sinnvoll, weil die Berechnung für die Optimierung eine vollständige Aufzählung aller möglichen Partitionen braucht. Stattdessen adoptieren die meisten Anwendungen bekannte Heuristik-Methoden, z.B. der k-Means-Algorithmus, der die Qualität des Clusteranalyse und die Annäherung eines lokalen Optimums zunehmend verbessern.

Hierarchische Methode

Durch diese Methode wird eine hierarchische Unterteilung des gegebenen Datasets erstellt. Eine hierarchische Methode kann entweder *agglomerativ* oder *divisiv* sein und basiert auf der Frage, wie die hierarchische Zersetzung geformt wird. Die *agglomerative* Methode wird auch *Bottom-up*-Methode genannt und in diesem Fall bildet jedes Datenobjekt am Anfang selbst eine separate Gruppe. Mithilfe dieser Methode werden separate Datenobjekte oder Gruppen zusammengefügt, bis alle Gruppen zu einer Gruppe zusammengefügt sind oder ein Stop-Kriterium als erfüllt gemeldet wird. Im Gegensatz dazu wird die divisive Methode auch als *Top-down*-Methode bezeichnet und in diesem Fall befinden sich alle Objekte am Anfang in einem gleichen Cluster. Durch die schrittweise Iteration wird ein großes Cluster zu mehreren kleinen Clustern zerlegt, bis sich jedes Objekt am Ende in einem separaten Cluster befindet oder eine Beendigungsbedingung erfüllt ist. Diese Methode kann entweder auf der Distanz oder auf der Dichte und der Kontinuität basieren.

Dichtebasierte Methode

Die meiste Partitions-Methoden bilden die Cluster für das Datenobjekt mithilfe der Distanzen zwischen den Objekten. Als Nachteil ist deutlich zu erkennen, dass nur kugelförmige Cluster gefunden werden können und bei der Untersuchung der Cluster mit einer beliebigen Form Schwierigkeiten aufkommen. Deshalb wurde eine andere Methode entwickelt, die auf der Grundlage des Gedankens der Dichte steht. Die generelle Idee dieser Methode ist, mit der Entwicklung eines gegebenen Clusters fortzufahren, sobald die Dichte von den benachbarten Datenpunkten einen Schwellenwert überschreitet. Mithilfe dieser Methode können verrauschte Daten und Ausreißer bereinigt werden und Cluster mit beliebigen Formen gefunden werden.

2.4.5 Angewendete Clusteranalyse-Algorithmen im Experiment

Weil der Datentyp von den Firmendaten nominal ist, sollen die Clusteralgorithmen an diesem Datentyp angepasst werden. Die Clusteranalyse für nominale Daten ist herausfordernd und schwieriger als die Clusteranalyse für numerische Daten, weil die meisten konventionellen Methoden der Clusteranalyse normalerweise für numerische Daten geeignet sind, z. B. die Berechnung der Distanz, die Bestimmung des Centroids und die Einschätzung der Dichte [Agg15, S. 206]. In diesem Abschnitt werden zwei Clusteralgorithmen vorgestellt, die auch für die nominalen Daten geeignet sind. Die notwendigen Modifikationen der traditionellen Clusteralgorithmen zur Erledigung der Anpassung an den nominalen Daten werden dabei im Detail behandelt.

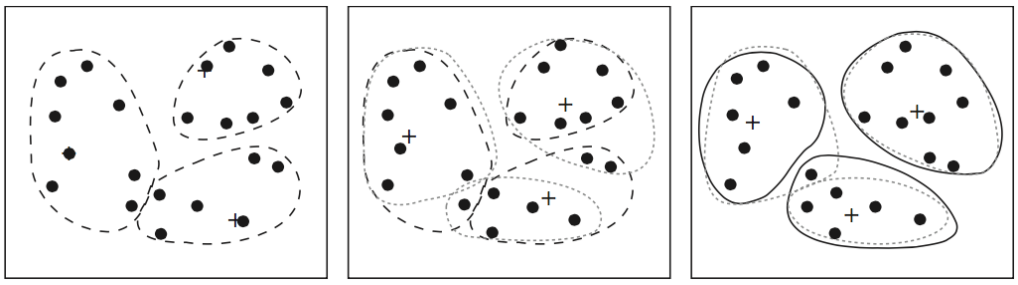
k-Means (Centroid-basierter Algorithmus)

In diesem Abschnitt wird zuerst die Funktionsweise dieses Algorithmus vorgestellt. Danach wird die Modifikation dieses Algorithmus zur Anwendung auf die nominalen Daten erläutert.

Funktionsweise

Der k-Means-Algorithmus definiert das Centroid eines Clusters als den Mittelwert der Punkte, die sich innerhalb dieses Clusters befinden. Zuerst werden k Datenobjekten im Dataset zufällig ausgewählt und anfänglich das Centroid jedes Clusters wird durch einen von den k Datenobjekten repräsentiert. Jedes von den verbleibenden Datenobjekten wird dem Cluster zugeordnet, mit dem es die meiste Ähnlichkeit hat. Der Zuordnungsprozess basiert auf dem euklidischen Abstand zwischen dem Datenobjekt und dem Centroid des entsprechenden Clusters. Danach erhöht der k-Means-Algorithmus iterativ die Variation zwischen unterschiedlichen Clustern. Das heißt: Jedes Cluster berechnet einen neuen Centroid durch die Datenobjekte, die in der letzten Iteration diesem Cluster zugeordnet wurden. Danach werden alle Datenobjekte nochmals zugeordnet unter Nutzung des im letzten Schritt aktualisierten Centroid, der als der neue Centroid des Clusters betrachtet wird. Diese Iterationen setzen sich bis zur Stabilität des Zuordnungsprozesses fort. Stabilität heißt, dass das Centroid des Clusters in dieser Iteration identisch zu dem Centroid von der letzten Iteration bleibt [HKP12, S. 452].

Die Abbildung 2.2 trägt visuell zum besseren Verständnis der Funktionsweise des K-Means-Algorithmus bei.



(a) Anfängliches Cluster (b) Iteration (c) Letztendliches Cluster

Abbildung 2.2: Funktionsweise des k-Means-Algorithmus (nach [HKP12, S. 453])

Der Centroid jedes Clusters wird mit dem Zeichen „+“ gekennzeichnet. Die drei Abbildungen zeigen genau die obere schriftliche Beschreibung der Funktionsweise des k-Means-Algorithmus.

Die Bestimmung der Summe von Clustern, also der Wert von k, kann mithilfe des k-Means-Algorithmus nicht realisiert werden, sondern der Benutzer entscheidet darüber. Der k-Means-Algorithmus ist normalerweise geeignet für numerischen Daten. Wenn der Datentyp von den Attributen nominal ist, sollen die Datenwerte in einen numerischen Datentyp transformiert werden [CL16, S. 143]. Im kommenden Abschnitt wird die notwendige Modifikation für die Anwendung des k-Means-Algorithmus auf die nominalen Daten erläutert.

k-Means-Algorithmus für die nominalen Daten

Nach [Agg15, S. 206f.] ist die Konvertierung der nominalen Daten zu binären Daten eine gute Lösung zur Durchführung der Clusteranalyse der nominalen Daten, weil die binären Daten eine spezielle Form der numerischen Daten sind und damit die Clusteralgorithmen für numerische Daten auf die konvertierten nominalen Daten angewendet werden können. Der genaue Transformationsprozess wurde schon in Abschnitt 2.3.2 erläutert und wird somit hier nicht nochmals wiederholt. Im späteren Experimentprozess wird der konkrete Anwendungsprozess mit den Experimentdaten genau gezeigt.

Nach der Erklärung in Abschnitt 2.4.3 ist die Basisfunktionsweise der Centroid-basierten Methode die wiederholte Festlegung eines Centroids von Clustern und die Festlegung der Ähnlichkeit zwischen den Centroids und den originalen Datenpunkten. Die allgemeinen Algorithmen der Centroid-basierten Methode funktionieren durch die iterative Festlegung der Centroids von Clustern und die Zuordnung der Datenpunkte zu dem nächstliegenden Centroid. In den höheren Ebenen bleiben solche Schritte gleich für die nominalen Daten. Aber die spezifizierten Schritte vom konventionellen k-Means-Algorithmus werden auch durch die Eigenschaft der nominalen Daten beeinflusst und nach [Agg15 S. 207f.] in folgenden zwei Aspekten modifiziert:

1. Centroid eines nominalen Datasets

Gemäß der Erklärung im letzten Paragraph ist es für den k-Means-Algorithmus notwendig, das Centroid eines Datasets festzulegen. Beim Fall von numerischen Daten wird dieses Centroid durch die Berechnung des Centroids festgelegt, während für die nominalen Daten das gleichwertige Centroid durch ein Wahrscheinlichkeitshistogramm repräsentiert wird. Für jedes Attribut *i* und den zugehörigen Attributwert v_j repräsentiert der Histogramm-Wert p_{ij} den Anteil der Datenobjekte eines Clusters am Attributwert v_j. Dieser Anteil wird auch als „Realtive Häufigkeit (RH)“ der Datenobjekten dieses Clusters am Attributwert v_j. geannt und dieser Begriff wird häufig auf die Interpretation der Experimentergebnisse im späteren Modellierungsprozess angewendet. Deshalb ist für ein Dataset mit *d* Dimensionen das Centroid eines Clusters eine Menge von *d* unterschiedlichen Histogrammen, die die Wahrscheinlichkeitsverteilung der nominalen Attributwerte jedes Attributs in diesem Cluster repräsentieren.

2. Berechnung der Ähnlichkeit zu Centroids

Das Ziel ist, die Ähnlichkeit zwischen einem Wahrscheinlichkeitshistogramm und dem nominalen Attributwert zu ermitteln. Die Wahrscheinlichkeit der einzelnen Attribute wird summiert zur Festlegung der totalen Ähnlichkeit. Jeder Datensatz wird einem Centroid mit der größten Ähnlichkeit zugeordnet.

Die anderen Schritte des k-Means-Algorithmus bleiben identisch zum Prozess für die numerischen Daten. Die Effizienz des k-Means-Algorithmus hängt hauptsächlich von der Verteilung

der Attributwerte ab. Wenn die Attributwerte schief verteilt sind, kann die histogrammbasierte Variation der anpassungsbasierten Maßnahmen schlecht durchgeführt werden, weil dabei jeder Attributwert als von gleichem Gewicht betrachtet wird. Eine gute Lösung für dieses Problem ist die Zuordnung der Gewichte für die unterschiedlichen nominalen Attributwerte. Diese Gewichte können sowohl die Erstellung des Wahrscheinlichkeitshistogramms als auch die Berechnung der anpassungsbasierten Ähnlichkeit beeinflussen [Agg15, S. 208]. Bei der „fachliche Kodierung“ im Abschnitt 4.1 werden die Attributwerte jedes Attributes nach der Höhe ihrer prozentualen Anteile aggregiert, damit jede Aggregationsgruppe ein gleiches Gewicht hat.

Erwartungsmaximierung (Wahrscheinlichkeitsmodellbasierter Algorithmus)

Identisch zu letzten Abschnitt wird zuerst die Funktionsweise dieses Algorithmus vorgestellt und danach werden die notwendigen Modifikationen für die Anwendung auf nominale Daten erläutert.

Funktionsweise

Der k-Means-Algorithmus ist ein harter Clusteralgorithmus und jeder Datenpunkt wird einem speziellen Cluster deterministisch zugeordnet. Der wahrscheinlichkeitsmodellbasierte Algorithmus ist ein weicher Algorithmus und jeder Datenpunkt hat eine „Nicht-null“-Zuordnungswahrscheinlichkeit zu vielen Clustern. Diese weiche Lösung kann durch die Zuordnung eines Datenpunktes zu einem Cluster auch zu einer harten Lösung konvertiert werden, wenn dieser Datenpunkt die größte Zuordnungswahrscheinlichkeit für dieses Cluster hat [Agg15, S. 173].

Der wahrscheinlichkeitsmodellbasierte Algorithmus funktioniert mithilfe eines mischungsbasierten Generativ-Modells. Es wird angenommen, dass die Daten der Mischung von k Verteilungen mit den Wahrscheinlichkeitsverteilungen $G_1 ..... G_k$ erzeugt werden. Jede Verteilung G_i repräsentiert ein Cluster und wird auch als eine Mischungskomponente betrachtet. Nach [Agg15, S. 173] wird jeder Datenpunkt $\overline{X_i}$, wo $i \in \{1....n\}$ mithilfe dieses mischungsbasierten Generativ-Modells durch folgende zwei Schritte erzeugt:

1. Auswahl einer Mischungskomponente mit der vorrangigen Wahrscheinlichkeit $\alpha_i = P(G_i)$ mit $i \in \{1....n\}$. Es wird angenommen, dass die Wahrscheinlichkeitsverteilung G_r ausgewählt wird.
2. Erzeugung eines Datenpunktes aus der Wahrscheinlichkeitsverteilung G_r.

Die Parameter der Verteilung jeder Komponente, wie beispielsweise der Mittelwert und die Varianz, sollen von den Daten eingeschätzt werden, damit die Daten die maximale Wahrscheinlichkeit haben, durch das Genarativ-Modell erzeugt zu werden. Dieser Prozess wird mithilfe des EM-Algorithmus realisiert. Die Parameter von unterschiedlichen Mischungskomponenten können zur Beschreibung von Clustern angewendet werden [Agg15, S. 174].

Der EM-Algorithmus beginnt mit der anfänglichen Einstellung der Parameter und danach werden Iterationen durchgeführt, bis es keine Optimierungsmöglichkeit der Cluster mehr gibt. Das heißt, dass die Konvergier- oder Änderungsmöglichkeit dann ausreichend klein ist [HKP12, S. 505]. Identisch zum k-Means-Algorithmus wird der EM-Algorithmus auch als Partitions-Methode erachtet. Bei der Clusteranalyse durch den EM-Algorithmus wird angenommen, dass die Daten mithilfe eines zufälligen Verfahrens erzeugt werden. Das genaue Cluster wird mithilfe der Gaußverteilung approximiert. Es wird angenommen, dass die Datenbereitstellung vom

EM-Algorithmus aus den k Gaußverteilungen erzeugt wird. Nach [CL16, S. 157] werden das Ziel und die Schritte des EM-Algorithmus folgendermaßen definiert:

„Das Ziel des EM-Algorithmus ist es daher, die k Gaußverteilungen zu finden, für die die Wahrscheinlichkeit, dass die gegebenen Daten aus ihnen entstanden sind, maximal ist.“

Damit ist die Funktionsweise des EM-Algorithmus ähnlich wie die vom k-Means-Algorithmus: Zuerst wird ein beliebiger Wert als Anfangswert ausgewählt und dieser Wert wird danach iterativ optimiert.

Nach [CL16, S. 158] wird der EM-Algorithmus in folgenden Schritten durchgeführt:

1. Der iterative Algorithmus berechnet zu den initialen Belegungen die Wahrscheinlichkeiten $P(x)$, $P\ (x_j \mid C_i)$ und $P\ (C_i \mid x_j\ )$.
2. Dann werden aus diesem Werten neue Mittelwerte der *k* Cluster bestimmt.
3. Dazu werden W_i, μ_{C_i} und Σ_{C_i} neu berechnet.
4. Aus diesen ergeben sich dann wiederum neue Wahrscheinlichkeiten.
5. Dies wird so lange wiederholt, bis *E* nicht mehr erhöht werden kann.

Dabei ist $P\ (x_j \mid C_i)$ die Wahrscheinlichkeit des Datenobjektes x_j und x_j gehört zum Cluster C_i. $P(x)$ ist die Gesamt-Wahrscheinlichkeitsdichte von Cluster C_i. W_i funktioniert als ein Gewichtungsfaktor und ist die relative Häufigkeit von Datenobjekten im Cluster C_i. E bedeutet den Erwartungswert und dient hier als ein Prüfungswerkzeug und durch dessen Berechnung wird überprüft, ob die Wahrscheinlichkeit, dass ein Datenobjekt sich in einem bestimmten Cluster befindet, maximal ist. Der *E*-Wert soll durch iterative Optimierungen maximiert werden. Die anderen Parameter sind nicht relevant für den späteren Experimentprozess und werden somit nicht weiter in dieser Masterarbeit erläutert. Nachfolgend werden die Formeln für die relevanten Parameter nach [CL16, S. 157] präsentiert.

$$P(x) = \sum_{i=1}^{k} W_i \cdot P\ (x \mid C_i) \qquad \textbf{(Formel 2.9)}$$

$$W_i = \frac{Anzahl\ der\ Objekte\ im\ Cluster\ C_i}{Anzahl\ aller\ Objekte} \qquad \textbf{(Formel 2.10)}$$

$$E = \sum_x \log(P(x)) \qquad \textbf{(Formel 2.11)}$$

EM-Algorithmus für die nominalen Daten

Ungleich zu der konventionellen Durchführung des k-Means-Algorithmus kann der EM-Algorithmus sowohl auf numerische als auch auf nominale Daten angewendet werden [Rap16]. Das Generativ-Modell kann für fast jeden Datentyp erzeugt werden, sobald eine geeignete Generierungs-Wahrscheinlichkeitsverteilung für jede Mischungskomponente definiert werden kann. Dieser Vorteil bringt hohe Flexibilität bei der Anwendung des wahrscheinlichkeitsbasierten Clusteralgorithmus auf die verschiedenen Datentypen [Agg15, S. 211].

Der erste Hauptunterschied im Vergleich mit dem numerischen Fall ist: Der weiche Zuordnungsprozess und der Einschätzungsprozess vom Parameter hängen von dem relevanten Wahrscheinlichkeitsverteilung-Modell für den entsprechenden Datentyp ab. Wenn k Mischungskom-

ponenten durch $G_1 ..... G_k$ bezeichnet werden, werden die Erzeugungsschritte des Generativ-Modells nach [Agg15, S. 211] zu folgenden zwei Schritten modifiziert:

1. Auswahl einer Mischungskomponente mit der vorrangigen Wahrscheinlichkeit α_i mit $i \in \{1....k\}$.
2. Wenn die *m*-te Mischungskomponente im ersten Schritt ausgewählt wird, wird danach ein Datenpunkt von G_m erzeugt.

Der zweite Hauptunterschied im Vergleich mit dem numerischen Fall liegt in der mathematischen Funktion vom generativen Modell für *m*-te Cluster (Mischungskomponente) G_m. Beim Fall von nominalen Daten ist die mathematische Funktion die diskrete Wahrscheinlichkeitsverteilung, während beim Fall von numerischen Daten die mathematische Funktion die Wahrscheinlichkeitsdichte Funktion ist. Es wird beim Fall von nominalen Daten angenommen, dass der *j*-te nominale Attributwert vom *i*-ten Attribut unabhängig durch die Mischungskomponente (Cluster) *m* mit der Wahrscheinlichkeit p_{ijm} erzeugt wird. Die mathematischen Funktionen von beiden Datentypen sind zwar unterschiedlich, aber die konkreten Formeln und die Berechnungsschritte sind gleich. Bei der Berechnung der Wahrscheinlichkeit p_{ijm} wird der Parameter α_m als durchschnittliche Zuordnungswahrscheinlichkeit der Datenpunkte eingeschätzt, die dem Cluster *m* zugeordnet werden [Agg15, S. 211f.]. Deshalb soll der Parameter „initial distribution" im RapidMiner vom späteren Experimentprozess mit der Option „average parameter" festgelegt werden.

2.4.6 Clustervalidierung

Nach der Durchführung einer Clusteranalyse ist es wichtig, die Qualität der Clusteranalyse zu validieren. Dieses Problem wird auch „Clustervalidierung" genannt. Die Durchführung der Clustervalidierung ist normalerweise schwierig in praktischen Datasets, weil sie mithilfe von *unsupervised*-Methoden realisiert wird. Deshalb stehen eigentlich keine genauen Clustervalidierungsmethoden zur Verfügung, die von der externen Seite bereitgestellt werden [Agg15, S. 195f.]. Von [TSK06, S. 553] werden die folgenden wichtigen Aspekte der Clustervalidierung benannt.

1. Festlegung der Summe von Clustern.
2. Validierung über das Anpassungsniveau der Clusteranalyse-Ergebnisse an den originalen Daten ohne Verweisung der externen Informationen.
3. Vergleich des Ergebnisses einer Clusteranalyse mit den externen bekannten Ergebnissen (ground truth), z. B. dem von der externen Seite bereitgestellten Klassen-Label.
4. Vergleich von zwei Clustern zur Festlegung des besseren Clusters.

Nach den oben gezeigten Aspekten unterteilen sich die Clustervalidierungsmethoden hauptsächlich in drei Kategorien, nämlich die internen, externen und relativen Clustervalidierungsmethoden. In diesem Abschnitt werden diese drei Kategorein nun nacheinander erläutert.

Interne Clustervalidierungsmethoden

Die internen Validierungsmethoden konzentrieren sich auf die Informationen, die das Cluster enthält, und beziehen sich auf die Frage, wie Datenpunkte unter Berücksichtigung von diesen Informationen aufgestellt werden. Ein gutes Clusteranalyseergebnis ist das Herausfinden von Clustern, bei denen sich die Datenpunkte innerhalb eines Clusters nahe beieinander befinden.

Die Kompaktheitsmaß-Methode beschäftigt sich mit diesem Thema. Das andere Merkmal von guten Clustern ist, dass alle Cluster wohl getrennt aufgestellt werden und dies wird auch bei den Separationsmaß-Methoden geprüft [HK14, S. 161f.].

Das Hauptproblem von den internen Clustervalidierungsmethoden ist, dass ihr Ergebnis zur Anpassung an den Algorithmen wahrscheinlich verzerrt wird. Bei den meisten Fällen sind die angewendeten Methoden zur Validierung der Qualität eines Algorithmus die objektiven Funktionen. Das verursacht deutliche Probleme, wenn der originale Algorithmus eine ungleichartige Methodik im Vergleich mit der Validierungsfunktion hat. Wenn der originale Algorithmus eine zur Validierungsfunktion ähnliche objektive Funktion hat, gibt es eine höhere Wahrscheinlichkeit, eine gute Note zu erhalten. Das heißt, dass die interne Validierungsfunktion versucht, nach einem Prototyp-Modell für das Algorithmus zu benoten. Das Ergebnis der internen Clustervalidierung zeigt nur die Anpassungsgenauigkeit des originalen Clusteralgorithmus an die Validierungsfunktion, nämlich das Prototyp-Modell. Der praktische Aspekt wird jedoch nicht ausreichend berücksichtigt [Agg15, S. 196f.]. Deshalb werden die internen Validierungsmethoden in dieser Masterarbeit nur als eine Referenz betrachtet, aber nicht die Hauptbeurteilungskriterien sein. Weil die dichtebasierte Clusteranalyse-Methode nicht geeignet für die nominalen Daten ist, wird die dichtebasierte Clustervalidierungsmethode in dieser Masterarbeit nicht behandelt. Nun werden die zwei internen Validierungsmethoden vorgestellt, die im späteren Experiment angewendet werden.

Davies-Bouldin-Index

Der Davies-Bouldin-Index ist eine Mess-Methode zur Einschätzung der optimalen Anzahl von Clustern in einem Dataset. Er wurde ursprünglich für den k-Means-Algorithmus definiert, weil der Cluster-Mittelpunkt, das Cluster-Centroid oder Cluster-Prototyp dieses Algorithmus eindeutig definiert werden. Bevor diese Methode angewendet wird, soll der k-Wert des Clusters vorher angenommen [GMJ13, S. 218].

Durch diese Methode werden alle Cluster nacheinander validiert. Die konkrete Funktionsweise dieser Methode ist: Für jedes Cluster wird ein anderes Cluster festgelegt und das gesuchte Cluster soll einen maximalen „Verhältniswert“ mit dem originalen Cluster haben. Dieser Verhältniswert wird durch die durchschnittliche Intracluster-Distanz von zwei Punkten, die zu beiden Clustern gehören, und die Distanz zwischen den beiden Clustern berechnet. Der Ergebniswert dieser Methode soll gering sein, wenn das entsprechende Cluster kompakt und weit getrennt ist. Mithilfe dieser Methode werden sowohl die Kompaktheit als auch die Trennbarkeit eines Clusters validiert und ein gutes Ergebnis dieser Methode zeigt ein gutes Cluster.
Formel 2.12 zeigt die Formel dieser Methode nach [Cic15, S. 382]:

$$db_{\delta,S}(d_1, d_2) = \frac{\Delta_{\delta,S}(d_1)+\Delta_{\delta,S}(d_2)}{\delta(\zeta_{d_1},\zeta_{d_2})} \qquad \textbf{(Formel 2.12)}$$

$$\Delta_{\delta,S}(d) = \frac{1}{|S^d|}\sum_{x\in S^d}\delta\,(x,\zeta_d)$$

wobei d_1, d_2 zwei Cluster bedeuten, δ das Unähnlichkeits-Maß ist, S das Dataset darstellt und $\Delta_{\delta,S}(\mathrm{d})$ die mittlere Unähnlichkeit zwischen den Datenpunkten von Cluster d und ihrem Mittelpunkt meint.

Wahrscheinlichkeitsmaß-Methode

Das Ziel dieser Methode ist, mithilfe eines Mischungs-Modells die Qualität einer speziellen Clusteranalyse einzuschätzen. Es wird angenommen, dass das Centroid jeder Mischungskomponente identisch zum Centroid jedes entdeckten Clusters ist. Die anderen Parameter jeder Mischungskomponente werden durch die Entdeckungsclusteranalyse mithilfe einer Methode berechnet, die ähnlich wie der Maximierungsschritt des EM-Algorithmus ist. Die „Log-Wahrscheinlichkeits-Maße" werden als Ergebnis erhalten. Das heißt, dass die Clustervalidierung beim EM-Algorithmus durch die Berechnung des E-Wertes realisiert wird [Agg15, S. 197]. Die genaue Formel zur Berechnung des E-Wertes wurde schon in Abschnitt 2.4.4 gezeigt und somit hier nicht wiederholt.

Externe Clustervalidierungsmethoden

Nach der Datenanalyse und der Fragestellung wird entschieden, dass die externen Clustervalidierungsmethoden im Experimentprozess dieser Masterarbeit nicht angewendet werden, weil die Firmendaten ungeordnet sind und keine Klassierer oder „ground truth" dafür vorhanden sind.

Relative Clustervalidierungs-Methoden

Durch die relativen Clustervalidierungs-Methoden können die Clusteranalyse-Ergebnisse, die durch zwei unterschiedliche Clusteranalyse-Algorithmen berechnet werden, nach bestimmten Kriterien verglichen werden, damit aus den beiden Clustern das Cluster mit der besseren Qualität ausgewählt werden kann [HK14, S. 163].

In Abschnitt 2.4.2 wurde bereits der Zusammenhang zwischen der Clusteranalyse und dem Data Mining-Verfahren erläutert. Die Clusteranalyse dient als ein Vorbereitungsschritt für die anderen DM-Verfahren. Zudem wird das Clusteranalyseverfahren mit dem Klassifikationsverfahren verglichen. Das Cluster wird normalerweise als eine implizite Klasse betrachtet und die Clusteranalyse wird auch automatische Klassifikation genannt, weil keine Lernrichtung vor der Clusteranalyse festgelegt werden soll. Deshalb funktionieren grundsätzlich alle mathematischen Verfahren, die geeignet für das Klassifikationsverfahren sind, auch bei den Clusteranalyse-Verfahren. Eine wichtige Aufgabe dieser Masterarbeit ist das Herausfinden von eventuell vorhandenen Clustern bei den Firmendaten. Das dient als ein Vorbereitungsschritt für die spätere Klassifikation der Firmendaten. Weil die Anzahl des k-Wertes nach dem Datenanalysebedarf schon festgelegt ist, kann das Verfahren zur Beurteilung der Qualität einer Klasse auch auf die Clusteranalyse angewendet werden.

Nach [CL16, S. 95] wird eine Methode zur Beurteilung der Klassifikationsleistung für die relative Clustervalidation ausgewählt, nämlich die „Fehlerrate". Das Ziel dieser Methode ist, den Anteil der Datensätze zu berechnen, die der falschen Klasse zugeordnet werden. Nach dem Datenanalysebedarf ist diese Methode geeignet zur Beurteilung der Qualität eines Clusters und wird somit eingesetzt. Die Formel 2.13 gilt nach [CL16, S. 95] für diese Methode:

$$Fehlerrate = \frac{die\ Anzahl\ der\ falsch\ klassifizierten\ Datensätze}{die\ Anzahl\ aller\ Datensätze} \quad \textbf{(Formel 2.13)}$$

Das Vergleichskriterium ist: Je geringer die Fehlerrate ist, desto besser ist die Qualität des Ergebnisses.

3. Anwendung der Datenvorverarbeitungs-Verfahren auf die Firmendaten

Im letzten Kapitel wurden die relevanten Theorien zum Data Mining-Prozess erläutert. Ab diesem Kapitel fängt nun der Praxisabschnitt an. Genau wie die Erklärungen im Abschnitt 2.2.1 wird der Experimentprozess nach dem Vorgehensmodell zur MESC von [ITP16] durchgeführt. Am Anfang jedes Abschnitts werden zuerst die entsprechenden Aufgaben nach diesem Vorgehensmodell angezeigt. Im Kapitel 3 werden die Datenvorverarbeitungsschritte, die im Abschnitt 2.3 ausführlich erläutert worden sind, auf die Firmendaten angewendet. Das heißt, dass der konkrete Modellierungsprozess der Datenvorverarbeitung in diesem Kapitel erläutert wird. Zuerst werden die Vorbereitungsaufgaben im Abschnitt 3.1 erledigt. Anschließend werden drei Datenvorverarbeitungsschritte im Abschnitt 3.2 ausführlich anhand der Experimentdaten erläutert. Danach wird das vollständige Experimentmodell im Abschnitt 3.3 angezeigt und erläutert. Nach der Erläuterung der Experimentprozesse werden die Ergebnisse der Datenvorverarbeitung mithilfe der sogenannten *Problemanalyse* interpretiert, wobei die Attributwerte, die sich eng auf die ausgewählten problematischen Attributwerte beziehen, mithilfe der Vergleichstabelle herausgefunden und interpretiert werden. Zum Schluss wird das Fazit dieses Kapitels aus den Experimentprozessen und Ergebnissen gezogen.

3.1 Vorbereitung des Experiments

Bevor das Experiment formell durchgeführt wird, sollen zuerst einige Vorbereitungsaufgaben erledigt werden. In diesem Abschnitt werden die ersten zwei Phasen des Vorgehensmodells MESC nach [ITP16] behandelt, die in der Tabelle 3.1 stehen. Zuerst wird die Aufgabendefinition des Experiments kurz vorgestellt, wobei das Ziel und die Fragestellung für diesen Abschnitt dargestellt werden. Im zweiten Abschnitt wird der Auswahlprozess aus den relevanten Datenbeständen nach zwei Aspekten erläutert, nämlich in Form der Datenbeschaffung und der Datenauswahl. Um die Zusammenhänge zwischen verschiedenen Datentabellen besser zu verstehen, wird ein Datenmodell mithilfe des ER-Modells (Entity Relationship-Modellierung) erstellt. Zum Schluss wird die Software RapidMiner kurz vorgestellt, die im Experiment verwendet wird.

Tabelle 3.1: Phase 1 und 2 des Vorgehensmodells zur Musterextraktion in SCs (nach [ITP16])

Phase	Schritte	Kurzbeschreibung
1. Aufgabendefinition	1.1 Bestimmung der Aufgabenstellung	Formulierung der Aufgabenstellung des Supply Chain Managements (SCM) unter Berücksichtigung von gegebenen Randbedingungen und Festlegung der Zielkriterien
2. Auswahl der relevanten Datenbestände	2.1 Datenbeschaffung	Bestimmung und Zugang zu den Datenquellen und den zugehörigen Datenbeständen gemäß Zieldefinition
	2.2 Datenauswahl	Auswahl der Datenbestände mittels Kontextwissen (für

		Def. siehe Bullinger et al. 2009) zwecks Datenreduktion

3.1.1 Aufgabendefinition

Die Experimentdaten stammen aus einer produktionslogistischen Firma. Es gibt insgesamt drei Datenbestände, die analysiert werden sollen. Die erste Experimentaufgabe ist die Auswahl der nützlichen Datentabellen und die Entwicklung eines Zielformats für die Datenbestände. Anschließend als die zweite Aufgabe sollen die Daten vorverarbeitet werden, so dass die original ungeordneten Firmendaten nach dem Datenvorverarbeitungsprozess für den späteren DM-Prozess geeignet sind. Zusätzlich sollen die Aggregationsstufe und die Erweiterungspotenziale während des Datenvorverarbeitungsprozesses nach dem Bedarf der Fragestellung festgelegt werden. Die Fragestellung dieser Masterarbeit ist das Herausfinden von potenziellen Clustern, die innerhalb der Firmendaten „versteckt" sind. Das heißt, es wird die mögliche Gruppeneinteilung der Firmendaten angestrebt. Damit wird dann die Anwendung der Clusteranalyse auf die Firmendaten als die dritte Aufgabe definiert. Nach den Datenvorverarbeitungs- und Clusteranalyse-Prozessen sollen die nützlichen Kenntnisse aus den Daten extrahiert und die Ergebnisse interpretiert werden. Zum Schluss soll das angewendete MESC-Modell nach der praktischen Verwertbarkeit bewertet werden.

3.1.2 Auswahl der relevanten Daten

Nach der Aufgabendefinition sollen die Datenbestände ausgewählt werden, die mithilfe des DM-Prozesses analysiert werden sollen. Nach dem Vorgehensmodell zur MESC von [ITP16] wird dieser Abschnitt in zwei folgende Teile gegliedert.

Datenbeschaffung

Die Daten, die analysiert werden sollen, werden von der Firma bereitgestellt und in der Software SQL so gespeichert, dass sie jederzeit nach Bedarf aus SQL exportiert werden können. Es gibt insgesamt drei Datenbestände, die analysiert werden sollen. Der Datenbestand „AESBig" enthält die Daten von einem inländischen Standort, während die Daten des Datenbestandes „AESSmall" aus einem ausländischen Standort stammen. Der dritte Datenbestand dient als die Validierung der Ergebnisse und enthält zahlreiche fehlende und verrauschte Daten. Am Anfang stehen innerhalb der Datenbestände „AESBig" und „AESSmall" 15 Datentabellen zur Verfügung, nämlich „Workpiece", „OperationProtocol", „OperationResultProtocol", „Parameter-Description", „Workplace", „Order", „Line", „Process" usw.

Datenauswahl

Nach der Datenbeschaffung sollen die Daten nach ihrer Nützlichkeit ausgewählt werden. Die Datenauswahl wird nach folgenden drei Aspekten durchgeführt.

Auswahl der Datenbestände

Durch die Datenanlyse von den drei Datenbeständen der Firmen werden folgende Ergebnisse erworben: Die enthaltenden Datentabellen und die konkrete Attribute jeder Datentabelle von den Datenbeständen „AESBig" und „AESSmall" gleich sind. Deshalb wird der Modellierungsprozess von Datenvorverarbeitung und Clusteranalyse sich auf den Datenbestand „AESBig" konzentriert. In dieser Masterarbeit werden die allgemeinen Modellierungsprozesse von den

Datenvorverarbeitung- und Clusteranalyse-Verfahren durch die Daten vom Datenbestand „AESBig“ dargestellt und die genauen Analysenverfahren vom Datenbestand „AESSmall“ sind analog damit und werden nicht nochmal in dieser Masterarbeit genau behandelt. Der dritte Datenbestand enthält zahlreiche fehlende Werte und die Daten davon sind nicht geeignet für den späteren DM-Prozess. Deshalb in dieser Masterarbeit werden die Daten von diesem Datenbestnad nicht analysiert.

Auswahl der Datentabelle

Nach der Aufgabenstellung sollen die unterschiedlichen Datentabellen zusammengeführt und ein Zielformat für den Datenbestand entwickelt werden. Deshalb werden nun die nützlichen Datentabellen zur Zusammenführung der Datenbestände gewählt. Nach der Datenanalyse wird festgestellt, dass die unterschiedlichen Datentabellen Zusammenhänge miteinander zeigen. Manche Datentabellen umfassen wenige Datensätze, die schon in einer großen Datentabelle integriert sind, z. B. „SystemId“ und „PlantId“. Nach der Analyse von unterschiedlichen Datentabellen wurde entschieden, das Experiment auf die vier Hauptdatentabellen zu konzentrieren, nämlich „Workpiece“, „OperationProtocol“, „OperationResultProtocol“, „ParameterDescription“. Um den DM-Prozess effizient durchführen zu können, werden die vier Haupttabelle in einer Hauptdatentabelle integriert. Der Integrationsprozess wird später in Abschnitt 3.2.1 genau erläutert. Weil die meisten anderen Datentabellen schon in den obengenannten vier Hauptabellen integriert sind, ist der Datenverlust wegen der Auswahl der Datentabellen kaum zu sehen. Um ein besseres Verständnis für den Datenbestand zu erhalten, wird mithilfe des ER-Modells ein Datenmodell erstellt. Im nächsten Abschnitt wird dieses ER-Modell kurz erläutert.

Auswahl der Daten

Weil die gesamte Summe der Datenzeilen nach der Datenintegration hoch ist, kann der DM-Prozess nicht mit den vollständigen Daten in RapidMiner durchgeführt werden. Deshalb werden in dieser Masterarbeit zwei Datenstichproben in der integrierten Hauptdatentabelle mithilfe der Software „SQL“ genommen, nämlich „Top 100.000“-Datenzeilen und „Last 100.000“- Datenzeilen. Weil die gesamte Summe der Attribute nach der Datenintegration zu hoch ist, wird das Attribut „Tag“ ausgewählt, auf das sich die Datenanalyse konzentrieren soll, um den DM-Prozess besser durchführen zu können.

3.1.3 Analyse der Datentabellen mithilfe eines ER-Modells

Das ER-Modell ist eines der bekanntesten Konstruktionsmodelle für Datenbestände. Seine Bekanntheit basiert auf den folgenden Vorteilen: einfache grafische Repräsentation, grafische Erweiterbarkeit von einer gegebenen Konstruktion und sichere Struktur. Eine große Stärke des ER-Modells ist seine streng hierarchische Struktur, die eine sichere Durchführung garantiert [Tha00, S. 3].

Neben der ERM gibt es noch drei wesentliche Datenmodelle, nämlich das Netzwerkmodell, das Relationsmodell und das Entity Set-Modell. Im Vergleich mit diesen verfügt die ERM über folgende Vorteile: Die ERM sieht natürlicher als die anderen Datenmodelle aus, weil es die Entität und die Relation auch in der Realität gibt. Sie hat manche wichtigen semantischen Informationen über die reale Welt im Modell integriert. Weiterhin erzielt die ERM einen hohen

Grad an Datenunabhängigkeit auf der Grundlage der Set-Theorie und der Relationstheorie [Che76, S. 9f].

Der Begriff „Entity“ steht für einen Gegenstand, der deutlich identifiziert werden kann. Eine spezielle Person, ein Unternehmen oder sogar ein Ereignis sind Beispiele für den Begriff „Entity“. Der Begriff „Relationship“ bedeutet eine Assoziation zwischen Entitäten, z. B. „Mutter–Tochter“ ist ein Relationstyp zwischen zwei „Person“-Entitäten. Die Informationen und Eigenschaften über eine Entität oder einen Relationstyp werden durch Beobachtung oder Messung erhalten und solche Informationen können mithilfe des Begriffs „Attribut“ dargestellt werden. Die genaue Darstellungsform ist „Attributwert“. Beispielsweise sind „10“, „blau“, „Mustermann“ verschiedene Attributwerte. Der Datenbestand eines Unternehmens enthält relevante Informationen von den Entitäten und Relationstypen, für die sich das Unternehmen interessiert. Die Entitäten oder Relationstypen können wahrscheinlich nicht den kompletten Datenbestand eines Unternehmens genau beschreiben, weil es unmöglich oder nicht notwendig ist, alle potenziell verfügbaren Informationen über die Entitäten und Relationstypen zu protokollieren [Che76, S. 10ff]. In der Abbildung 3.1.1 wird ein einfaches Beispiel für die ERM gezeigt.

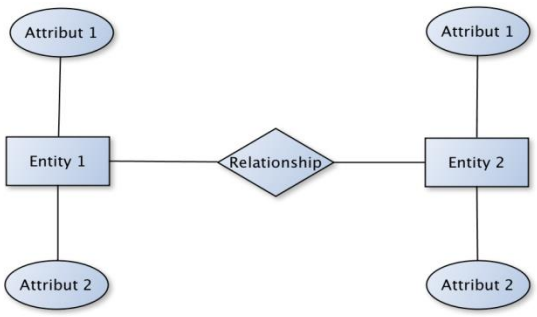


Abbildung 3.1: Beispiel des ER-Modelles

In der Abbidldung 3.1 gezeigten Beispiel eines ER-Modells werden Entität 1 und Entität 2 mithilfe eines Relationstyps miteinander verbunden. Jede Entität umfasst jeweils zwei unterschiedliche Attribute. Um die Relationen zwischen verschiedenen Datentabellen der Firmendaten übersichtlich zu beschreiben, wird ein ER-Modell dafür erstellt. Dieses ER-Modell wird zur Gewährleistung der Abbildungsqulität exportiert und in der begleitenden CD gespeichert.

Jetzt wird die Funktionsweise dieses ER-Modelles genau erläutert. Die Datentabellen „Workpiece“, „OperationProtocol“, „OperationResultProtocol“, „ParameterDescription“, „TraceabilityData“, „LineID“, „Product“, „Order“, „Workplace“ und „Process“ werden als Entitäten definiert und die entsprechenden Attribute der jeweiligen Datentabellen werden auch als Attribute von den entsprechenden Entitäten im ER-Modell eingefügt. Die unterschiedlichen Entitäten werden mithilfe der Relationstypen verknüpft, die die Relationen zwischen zwei nebeneinanderstehenden Entitäten darstellen. Der Relationstyp zwischen zwei nebeneinanderstehenden Entitäten beschreibt die Relation sowohl in schriftlicher Form als auch in einer Proportionsform. Das heißt: Jede Entität befindet sich in einer individuellen Ebene oder Reihenfolge im ER-Modell. Die Entität auf der ersten Ebene oder in oberster Reihe des ER-Modells steht auf der ganz oberen Seite des ER-Modells. Je höher die Ebene oder Reihenfolge ist, an desto niedrigerer Position befindet die Entität im ER-Modell. Jede Entität kann nach einem Attribut von ihr in der nächsten Ebene entweder spezialisiert oder generalisiert werden. Im dritten Fall ver-

fügen zwei Entitäten über ein gleiches Attribut und die Entität wird direkt nach diesem gleichen Attribut in der Entität von der nächsten Ebene integriert. Beim Fall „Spezialisierung“ wird das Proportionszeichen (1:N) neben dem Relationstyp eingefügt, während das Proportionszeichen (N:1) beim Fall „Generalisierung“ eingefügt wird. Beim Fall „direkte Verknüpfung“ wird das Proportionszeichen (1:1) angewendet. Jede Entität umfasst zahlreiche Attribute und nur die wesentlichen Attribute werden im ER-Modell eingefügt. Das Attribut, das die Entität eindeutig identifiziert, wird als identifizierender Attributwert unterstrichen.

Nun wird das ER-Modell der Firmendaten nach den obengenannten Theorien genau erläutert. Im Anhang 2 werden Screenshots der 10 Datentabellen gezeigt, die im ER-Modell als Entitäten stehen. Nach der Analyse der Datentabellen der Firma werden folgende Ergebnisse herausgefunden:

1. Die Datentabellen „Workpiece“, „OperationProtocl“ und „OperationResultProtocol“ verfügen über ein gleiches Attribut „WorkpieceGuid“. Bei der Datentabelle „Workpiece“ heißt dieses Attribut „Guid“, aber die Datenwerte sind genau gleich. Deshalb können diese drei Datentabellen mithilfe des Attributs „WorkpieceGuid“ integriert werden. Nach dem Attribut „WorkSequence“ kann die Datentabelle „Workpiece“ zur Datentabelle „OperationResult“ spezialisiert werden. Das heißt, dass der einzelne Attributwert vom Attribut „WorkpieceGuid“ nach den unterschiedlichen Attributwerten des Attributs „WorkSequence“ unterteilt wird. Deshalb verfügt die Datentabelle „OperationProtocol“ über mehr Datensätze als die Datentabelle „Workpiece“, weil jeder Datensatz der hinteren Datentabelle in mehrere Datensätze in der vorderen Datentabelle spezialisiert wird. Nach dem gleichen Verfahren wird die Datentabelle „OperationProtocol“ nach dem Attribut „ResultSequence“ in der Datentabelle „OperationResultProtocol“ spezialisiert. Bei den obengenannten beiden Spezialisierungsfällen wird das Proportionszeichen mit (1:N) markiert.

2. Da die Datentabellen „OperationResultProtocol“ und „ParameterDescription“ über ein gleiches Attribut „ParameterDescriptionId“ verfügen, werden beide mithilfe dieses Attributs integriert und das Proportionszeichen wird mit (1:1) eingegeben.

3. Die Datentabelle „Workpiece“ verfügt über die Attribute „ProductId“, „Line“, und „OrderId“ und diese drei Attribute haben eigene individuelle Datentabellen. Deshalb kann die Datentabelle „Workpiece“ nach den obengenannten drei Attributen in die Datentabellen „Product“, „Order“ und „Line ID“ generalisiert werden. Der Generalisierungsprozess wird mithilfe eines Beispiels vom Attribut „LineId“ und der entsprechenden Entität „Line ID“ genau erklärt: In diesem Fall werden zahlreiche Datensätze vom Attribut „LineId“ der Datentabelle „Workpiece“ in drei unterschiedliche Line ID-Nummern in der Datentabelle „Line“ zusammengefasst, nämlich Line 2, Line 3 und Line 4. Das Proportionszeichen wird mit (N:1) markiert. Die zwei anderen Attribute „ProductId“ und „OrderId“ werden nach dem gleichen Verfahren zu den Datentabellen „Product“ und „Order“ generalisiert.

4. Die Datentabelle „TraceabilityData“ wird als Validierungsdaten betrachtet. Am Ende des Experiments werden die Daten dieser Datentabelle helfen, die Ergebnisse zu validieren.

5. Jede Entität oder Datentabelle verfügt über zahlreiche Attribute. Aber die Attributwerte von vielen Attributen sind fehlende Werte und/oder schwer zu analysieren. Nach der groben Datenanalyse werden einige Attribute ausgewählt und als wesentlicher Attributwert von der jeweiligen Entität im ER-Modell unterstrichen.

3.1.4 Angewendete Data Mining-Software

Die Software, die im Experiment angewendet wird, heißt RapidMiner. In diesem Abschnitt werden das Aussehen und die üblichen Funktionen dieser Software kurz eingeführt.

RapidMiner ist eine Open Source-Plattform für Data Mining, wurde entwickelt und wird verwaltet von der RapidMiner GmbH. Früher wurde diese Software bekannt unter dem Namen „YALE (Yet Another Learning Environment) und wurde entwickelt an der Universität Dortmund. RapMin ist eine GUI (Graphical User Interface)-basierte Software, bei der der Arbeitsablauf eines DM-Prozesses entwickelt und aufgestellt werden kann [KD15, S. 371].

Aussehen

Im Anhang 1 wird das Aussehen der Software RapidMiner mithilfe der Beispieldaten und Beispielprozesse durch das Screenshot vorgestellt, die in der Software als Tutorium dienen. Nun werden die wesentlichen Funktionseinheiten, die im Screenshot stehen, kurz erläutert.

Repository

Diese Funktionseinheit steht auf der oben linken Seite des Screenshots. Repository ist ein Ort, an dem Benutzer ihre Daten, Prozesse und Modelle speichern und organisieren können. Deshalb wird das Repository als ein zentraler Platz für die Daten und Analyseprozesse betrachtet. Im Repository können die Ordner und die Unterordner zur Speicherung der Daten, Prozesse und Modelle erstellt werden [KD15, S. 375].

Operators

Diese Funktionseinheit steht auf der linken Seite des Screenshots. Jeder Operator ist ein Teil der Funktionalität, mit dem eine bestimmte Aufgabe durchgeführt wird, z. B. Feature Selection, Ersetzung von fehlenden Werten usw. Zahlreiche Operatoren stehen hier zur Verfügung und werden verschiedenen Kategorien zugeordnet, z. B. Data Access und Cleaning. Weil RapMin eine GUI-Software ist, bedeutet das Einfügen eines Operators auch die Addition einer Menge von Programmierungskodes zur Software, damit die Operatoren in RapidMiner nur visuelle Zeichen sind, die eine Menge von Programmierungskodes repräsentieren [KD15, S. 376f].

Problem

Diese Funktionseinheit steht auf der unteren linken Seite des Screenshots. Wenn der Prozess Probleme bei seiner Durchführung hat – entweder durch falsche Einstellung des Parameters oder falsche Auswahl des Operators –, wird die Fehlermeldung hier schriftlich angezeigt, damit der Benutzer die groben Informationen über die Fehler schnell erfahren kann.

Process

Diese Funktionseinheit steht genau in der Mitte des Screenshots. Ein einzelner Operator kann den DM-Prozess nicht alleine durchführen. Alle DM-Prozesse brauchen eine Serie von Kalkulations- und logischen Operationen. Ein typischer DM-Prozessablauf enthält folgenden Schritte:

1. Import der notwendigen Daten
2. Durchführung der Datenvorverarbeitung
3. Durchführung eines Modells mit Trainingsdaten
4. Validierung des Modells, um die Leistung der verschiedenen Modelle einzustufen
5. Anwendung des Modells zur Extraktion von Kenntnissen

Diese fünf Schritte können durch die Kombination von unterschiedlichen notwendigen Operatoren durchgeführt werden und jeder Operator hat seine individuelle und spezielle Aufgabe. Wenn

die notwendigen Operatoren je nach Bedarf miteinander kombiniert werden, wird ein Prozess aufgebaut. Zusätzlich können Notizen im Prozess eingefügt werden zum besseren Verständnis des einzelnen Operators, z. B. die gelbe Notiz im Screenshot.

Parameter

Diese Funktionseinheit steht auf der rechten Seite des Screenshots. Dort werden die wichtigen Parameter angezeigt und ihre Kennzahlen können hier beliebig festgelegt werden, z. B. die Kennzahl „Sample Size“ im Screenshot. Damit kann die Größe der Stichprobe festgelegt werden.

Help

Diese Funktionseinheit steht unten auf der rechten Seite des Screenshots. Hier werden nützliche Hilfeinformation, z. B. die Beschreibung der Funktionsweise, die Form von Input- sowie Output-Daten und die Erklärung der wichtigen Parameter des Operators angezeigt.

Übliche Funktionen

Neben den zahlreichen nützlichen Operatoren bietet RapidMiner auch umfangreiche statistische Funktion sowie Visualisierungsfunktionen zur Beschreibung von Daten. Identisch zum letzten Abschnitt „Aussehen“ wird nun wieder eine Beispieldatentabelle, „Iris“, ausgewählt, um die üblichen Funktionen der Software zu beschreiben. Im Anhang 1 werden das Screenshot von den Menüs „Datenübersicht“, „Datenstatistik“ und ein Datenvisualisierungsbeispiel angezeigt. Darunter werden jede Funktion kurz vorgestellt.

Datenüberblick

Mittels dieser Funktion werden die originalen Daten der Datentabelle angezeigt und unterschiedlichen Rollen, wie beispielsweise, „id“ und „label“, zugeordnet.

Datenstatistik

Mittels dieser Funktion werden die statistischen Informationen von den Attributen der Datentabelle angezeigt. Im Vergleich mit der Funktion „Datenüberblick“ sehen die Daten bei dieser Funktion übersichtlicher aus. Durch diese Funktion kann der Benutzer die wichtigen Eigenschaften der Daten von der Datentabelle schnell kennenlernen.

Datenvisualisierung

Mithilfe dieser Funktion können die Daten der Datentabelle grafisch dargestellt werden. Die Benutzer können die Gruppierungsspalten, Wertspalten und Aggregationsmethode nach eigenem Bedarf beliebig einstellen. Das Aussehen der Grafik lässt sich auch mithilfe der Einstellung bequem ändern, z. B. horizontale oder senkrechte Grafik. Im RapidMiner stehen viele Visualisierungsmethoden zur Verfügung, z. B. Balkendiagrammgrafik, Scatter, Pie, Series usw.

3.2 Aufbau des Experimentmodells

Ab diesem Abschnitt beginnt die Beschreibung des eigentlichen Datenvorverarbeitungsprozesses. Nach dem Vorgehensmodell von MESC [ITP16] sollen folgende Schritte bearbeitet werden.

Tabelle 3.2: Phase 3 des Vorgehensmodells MESC (nach [ITP16])

Phase	Schritte	Kurzbeschreibung
3. Daten-vorverarbeitung	3.1 Format-Standardisierung	Überführung der selektierten Datenbestände in ein Standardformat
	3.2 Gruppierung	Fachliche Gruppierung der Datenbestände unter Berücksichtigung der Aufgabenstellung
	3.3 Datenanreicherung	Datenanreicherung unter Einbeziehung von Kontextwissen
	3.4 Transformation	Prüfung auf Atomarität der Attribute, Anreicherung von Daten unter Zuhilfenahme von Kontextwissen, Merkmalsreduktion, Behandlung von fehlenden und fehlerhaften Merkmalen sowie Ausreißerkorrektur

Am Anfang werden die im ersten Abschnitt ausgewählten vier Hauptdatentabellen geprüft, ob deren Format zum Standardformat passt. Wenn nicht, sollen die Datentabellen in ein Standardformat überführt werden. Der zweite Schritt „*Gruppierung*“ wird in der Datenvorverarbeitung auch als „*Datenintegration*“ benannt. Bei diesem Schritt werden die vier Hauptdatentabellen nach der Aufgabenstellung integriert, um die Datenvorverarbeitung einheitlich besser durchführen zu können. Im dritten Schritt „*Datenanreicherung*“ wird nach der Überlegung des Kontextwissens geprüft, ob es notwendig ist, das aktuelle Dataset durch die neuen externen Attribute zu erweitern oder die bestehenden Attribute zu kombinieren zur Erstellung von neuen Attributen. Anschließend soll die Kompressionsmöglichkeit von bestehenden Attributen überlegt werden. Zum Schluss kommt der zentrale Schritt der Datenvorverarbeitung, die „*Transformation*“. Da die Datenvorverarbeitung mehrere Schritte enthält, wird dieser Schritt in drei Methoden unterteilt, die in Kapitel 2 schon erläutert worden sind, nämlich *Datenaggregation*, *Datenhomogenisierung* und *Feature Selection*. Die ersten drei Schritte des MESC-Modells werden ebenfalls diesen drei Verfahren zugeordnet. Die Format-Standardisierung wird der „*Datenhomogenisierung*“ zugeordnet und die zweiten und dritten Schritte der „*Datenaggregation*“. Das komplette Experimentmodell wird im nächsten Abschnitt vorgestellt, das aus vielen kleinen Teilen besteht. Um das Experimentmodell besser erklären zu können, wird das komplette Modell nach seinen unterschiedlichen Funktionalitäten in einzelne Modelle zerlegt. Dabei wird der Datenvorverarbeitungsprozess nach dem oben genannten Verfahren Schritt für Schritt erläutert. Die Namen der Parameter von den unterschiedlichen Operatoren im RapidMiner werden in der Beschreibung kursiv kenngezeichnet.

3.2.1 Datenaggregation

Durch die Datenanalyse wird herausgefunden, dass die originalen Datenbestände viele unterschiedliche Datentabellen enthalten und die großen Datentabellen zahlreiche Attribute und Datensätze haben. Aber die meisten Daten in den Datenbeständen sind redundant und nicht geeignet für den späteren DM-Prozess. Deshalb ist es notwendig, eine Datenaggregation vor dem DM-Prozess durchzuführen. In diesem Abschnitt werden die Experimentprozesse des Datenvorverarbeitungs- schritts „Datenaggregation“ nach dem Vorgehensmodell zur MESC von

[ITP16] sowie die Methoden, die im Abschnitt 2.3.3 vorgestellt wurden, mit den Experimentdaten Schritt für Schritt erläutert. Zuerst wird die Schritte der „Datenintegration“ vorgestellt. Anschließend werden die unterschiedlichen Methoden von „Datenanreicherung“ erläutert.

Gruppierung durch die Datenintegration

Nach der Erläuterung in Abschnitt 2.3.3 wird der Datenintegrationsprozess in drei Schritte unterteilt, nämlich Entitäten-Identifikationsproblem, Redundanz- und Korrelationsanalyse und Tupel-Duplikation. Der Datenintegrationsprozess im Modellierungsprozess wird auch nach diesen drei Schritten durchgeführt.

Entitäten-Identifikationsproblem
Nach der Aufgabenstellung sollen die Datenbestände aus verschiedenen Quellsystemen zusammengeführt werden. Deshalb wird eine „*fachliche Gruppierung*“ von den vier Haupttabellen durchgeführt, damit diese im späteren DM-Prozess gemeinsam analysiert werden können. Anhand der Analysenergebnisse des ER-Modells in Abschnitt 3.1.3 wird herausgefunden, dass Zusammenhänge zwischen den vier Haupttabellen bestehen. Das heißt, dass die vier Haupttabellen nach den zusammenhängenden Attributen zu einer Datentabelle integriert werden können. Der genaue Gruppierungsprozess wird mithilfe der Software SQL realisiert. In dieser Masterarbeit werden insgesamt 10 Datenstichproben genommen, wie bereits im Abschnitt 3.1.2 erläutert. In Anhang 3 wird die Software SQL mithilfe eines Screenshots präsentiert und die angewendeten Befehle werden im Anhang 4 vorgestellt.

Redundanz- und Korrelationsanalyse
Nach der Datenintegration beträgt die Attributsumme insgesamt 100. Diese Summe ist zu hoch und eine Datentabelle mit 100 Attributen ist nicht geeignet für den späteren DM-Prozess. Aber nach der Datenanalyse wird herausgefunden, dass viele Attribute Probleme zeigen und nicht geeignet für den späteren DM-Prozess sind. Solche Attribute werden als nutzlos betrachtet und können direkt manuell reduziert werden. Alle Problemattribute der HDT werden in der Tabelle 3.3 aufgelistet und entsprechend unterschiedlichen Problembeschreibungen sortiert.

Tabelle 3.3: Sortierung von Problemattributen

Problem-beschreibung	Problemattribute	Behandlungs-methode
Zahlreiche fehlende Werte	Version1, Version2, Version3, Version4, Version5, Unit, Description2, Format, DefaultValue	Filtern
Alle Werte sind null oder Sonderzeichen, zum Beispiel: N, NONE, NULL, ???	CarrierId, CarrierPosition, ProductVersion, TTableId, SerialNumber2, SerialNumber3, SerialNumber4, SerialNumber5, RoutingVersion, DataTransferState, Shift, ResultCode, ParameterDescriptionVersion, ParameterVersion, ParentParameterVersion, Binary, ParentParameterDescriptionVersion, LowerLimit, UpperLimit, IsStructMember	Filtern
Nur eine Wertkategorie	PlantId, SystemId, RoutingId, IndentLevel, ClassAssociatedTo, Attribute, IsActive, IsProtected, CreatedName, ChangedName	Filtern
Direkte Duplikate	WorkpieceGuid, RoutingId_1, RoutingVersion_1,	Filtern

	PlantId_1, SystemId_1, LineId_1, CarrierId_1, DataTransferState_1, TimeStamp_1, WorkpieceGuid_1, WorkSequence_1, RoutingId_2, RoutingVersion_2, RoutingSequence_1, Result_1, DataTransferState_2, TimeStamp_2, ParameterDescriptionID_1, IndentLevel_1, PlantId_2, SystemId_2, LineId_2, Type_1, Unit_1, TimeStamp_3	
Indirekte Duplikate	TimeStamp, BeginOfWork, EndOfWork, Result	Filtern
Zu viele verrauschte Daten	Value	Filtern

Nach der groben Sortierung der Problemattribute wird das einzelne Problem durch Beispiele von Problemattributen genau erläutert. Beim Experiment der Datenvorverarbeitung werden 100.000 Datenzeilen als Stichprobe als Dateninput genommen.

i. „Zu viele fehlende Werte“ und „Alle Werte sind null oder Sonderzeichen“
Die Abbildung 3.2 zeigt die Problemattribute, die zu viele fehlende Datenwerte haben oder bei denen alle Datenwerte null sind.

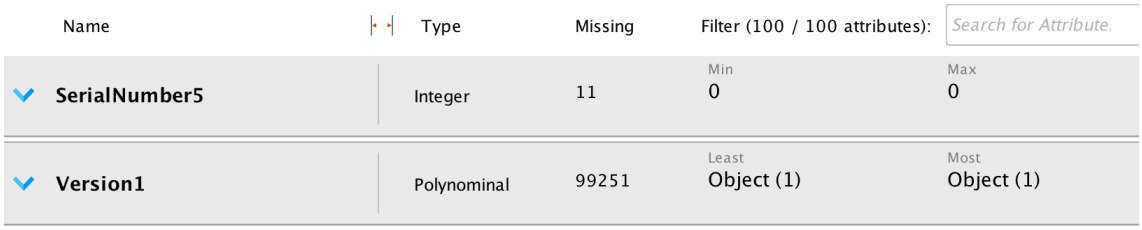


Abbildung 3.2: Beispiel von den Problemattributen (1) (nach RapidMiner)

Alle Datenwerte vom ersten Attribut sind null und fast alle Datenwerte vom letzten Attribut sind fehlende Werte. Solche Attribute enthalten fast keine nützlichen Daten und werden als redundant betrachtet.

Die Abbildung 3.3 zeigt ein Screenshot eines Attributs, dessen Werte fast alle Sonderzeichen sind.

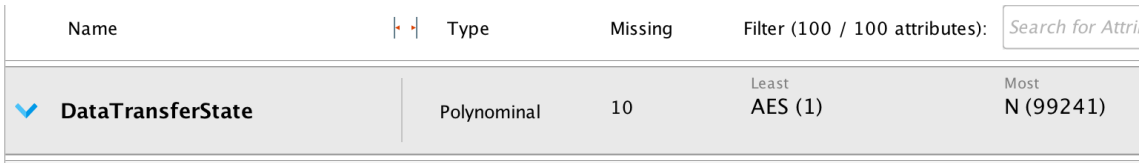


Abbildung 3.3: Beispiel von den Problemattributen (2) (nach RapidMiner)

In dieser Masterarbeit werden Zeichen wie z. B. „N“, „NONE“, „NULL“, „???“ als Sonderzeichen betrachtet. Der Attributwert vom oben gezeigten Attribut sind fast alle Sonderzeichen „N“ und können nicht im späteren DM-Prozess analysiert werden. Deshalb werden solche Attribute als nutzlos betrachtet

.

ii. „Nur ein Attributwert“
Die Abbildung 3.4 zeigt ein Attribut, das nur einen Attributwert hat.

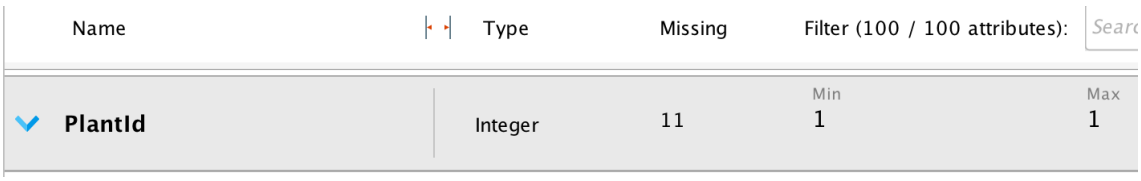


Abbildung 3.4: Beispiel von den Problemattributen (3) (nach RapidMiner)

Es wird herausgefunden, dass dieses Attribut nur den Wert 1 hat. Das heißt, dass das angezeigte Attribut nur einen Attributwert hat und nicht sinnvoll für den späteren DM-Prozess ist. Deshalb wird dieses Attribut als redundant betrachtet.

iii. Direkte Duplikat-Attribute
Im Anschluss werden zwei Attribute durch die Abbildung 3.5 angezeigt, die direkte Duplikate von anderen Attributen sind.

Abbildung 3.5: Beispiel von den Problemattributen (4) (nach RapidMiner)

In der Abbildung 3.5 gezeigten zwei Attributen sind Duplikate von den existierenden Attributen „LineId“ und „WorkSequence“ und werden als redundant betrachtet.

iv. Indirekte Duplikat-Attribute
In der Abbildung 3.6 werden drei Attribute präsentiert, die indirekte Duplikate von anderen Attributen sind.

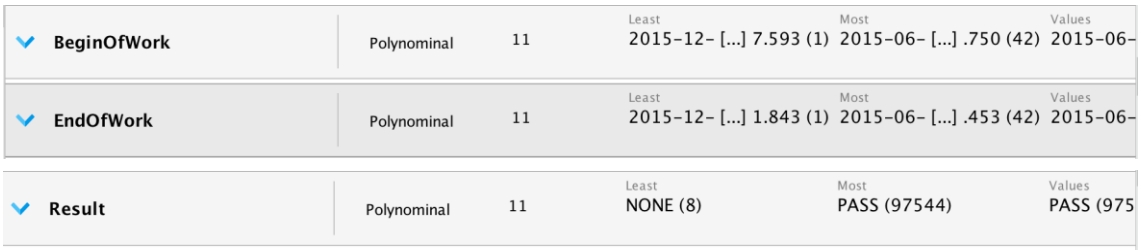


Abbildung 3.6: Beispiel von den Problemattributen (5) (nach RapidMiner)

Direkte Duplikat-Attribute sind einfache Kopien von existierenden Attributen. Die Attributwerte sind genau identisch zu der von den originalen Attributen und der Benutzer braucht sie nicht noch einmal zu analysieren. Bei den indirekten Duplikat-Attributen ist das etwas anders. Die Attributwerte bei diesem Fall sind zwar nicht identisch zu den von den originalen Attributen, aber die Datenwerte der indirekten Duplikat-Attribute hängen von den originalen Attributen ab. Das heißt, dass die indirekten Duplikat-Daten in dieser Masterarbeit die spezialisierte Version von den originalen Attributen sind, z. B. das Analyseergebnis des ER-Modells zeigt, dass die Datentabelle „Workpiece“ nach den Attributwerten vom Attribut „WorkSequence“ zur Datentabelle „OperationProtocol“ spezialisiert wird. Die Attribute „BeginOfWork“ und „EndOfWork“

gehören zur Datentabelle „OperationProtocol“ und sind die spezialisierte Version von den Attributen „BeginOfManufacturingTime“ und „EndOfManufacturingTime“, die zur Datentabelle „Workpiece“ gehören. Das heißt, dass die Analyse der Attribute „BeginOfManufacturingTime“ und „EndOfManufacturingTime“ schon ausreichend in dem späteren DM-Prozess vertreten sind und solche indirekten Duplikate als redundant betrachtet werden können.

Die Redundanzattribute werden im nächsten Abschnitt „Datenanreicherung“ manuell bereinigt. Beim Datenvorverarbeitungsprozess wird für die übrigen Attribute eine sogenannte „Korrelationsanalyse“ durchgeführt. Genau wie die Erläuterung in Kapitel 2 wird die Korrelationsanalyse mithilfe der Chi-Square-Statistik durchgeführt, die im Abschnitt 3.2.3 genau vorgestellt wird.

Tupel-Duplikate

Nach der Datenintegration werden zahlreiche Tupel-Duplikate erstellt, z. B. das Tupel (Tag, LineId). Der Grund dafür ist der unterschiedliche Spezialisierungsgrad der vier integrierten Datentabellen. Die relevanten Inhalte über den Spezialisierungsgrad wurden schon in Abschnitt 3.1.3 mithilfe des ER-Modells genau erläutert und werden somit hier nicht wiederholt. Weil die vier Datentabellen zusammen integriert werden und unterschiedliche Spezialisierungsgrade haben, können die Tupel-Duplikate in dieser Masterarbeit nicht einfach bereinigt werden, um die Vollständigkeit der Hauptdatentabelle nicht zu zerstören.

Datenanreicherung

Das Thema „Datenanreicherung“ bedeutet die Verstärkung des Datenbestands und umfasst hauptsächlich zwei Aspekte, nämlich das Einfügen der neuen Attribute von einer externen Seite und die Komprimierung des bestehenden Datenbestands. In Abschnitt 2.3.3 wurden die Notwendigkeit und die Funktionsweise des ersten Anreicherungsaspektes schon erläutert. Weil die Analysedaten dieser Masterarbeit aus einer echten Produktionslinie stammen und nicht ähnlich wie die Marktanalysedaten sind, ist es nicht notwendig, zusätzliche Daten von externer Seite einzufügen. Deshalb wird die Datenanreicherung in dieser Masterarbeit durch die Kompression der bestehenden Datentabelle durchgeführt und wird in folgenden Methoden spezialisiert:

1. Bereinigung der Redundanzattribute
2. Identifikation der Attribute, die zwar nützlich sind, aber gespart werden können
3. Kombination der bestehenden Attribute unter Zuhilfenahme von Kontextwissen (Attribut-Extraktion und Attribut-Konstruktion)
4. Diskretisierung

Nun werden die oben genannten vier Methoden anhand der Experimentdaten erläutert.

Bereinigung der Redundanzattribute

Die im letzten Abschnitt ausgewählten Redundanzattribute werden mithilfe des Operators „Select Attributes“ bereinigt. In der Abbildung 3.7 wird dieser Experimentprozess angezeigt.

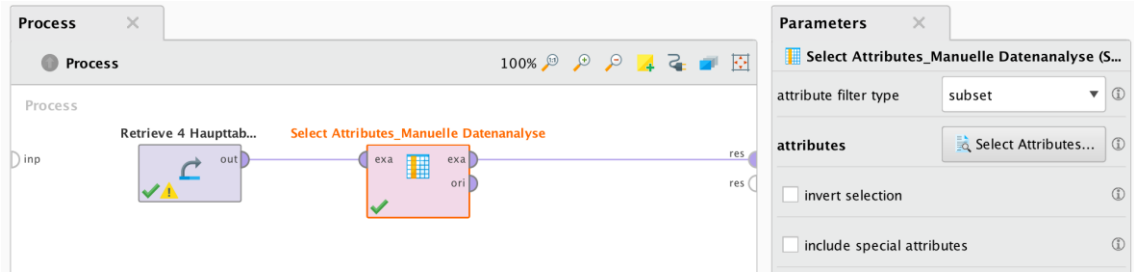


Abbildung 3.7: Modellprozess der Bereinigung der Redundanzattribute (nach RapidMiner)

Der Operator „Retrieve“ bedeutet eine Dateneingabe. Der Parameter *attribute filter type* soll als die Option *subset* festgelegt werden, damit mehrere Attribute gleichzeitig ausgewählt werden können. Mithilfe dieses Operators werden die nützlichen Attribute ausgewählt und um die redundanten Attribute wird bereinigt. Die genauen Einstellungen des Parameters „Select Attributes“ und die Ergebnisanzeige werden in Anhang 5 präsentiert.

Identifikation der Attribute, die zwar nützlich sind, aber gespart werden können

Durch die Datenanalyse wird herausgefunden, dass im Vergleich mit den anderen drei Datentabellen die Datentabelle „ParameterDescription“ den höchsten Spezialisierungsgrad hat und die Attribute dieser Datentabelle für die Beschreibung der Eigenschaften der verschiedenen ParameterDescriptionID (ParaDesID) dienen. Das heißt, dass die nützlichen Attribute „ClassAssociatedTo“, „Category“, „Name“, „Description“, „Type“, „DocRequired“ nur Zusammenhänge mit dem Hauptattribut „ParaDesID“ haben. Deshalb soll nur das Hauptattribut „ParaDesID“ bei der Datenanalyse berücksichtigt werden und die anderen sechs nützlichen Attribute können gespart werden, weil die die Attributwerte von anderen Attributen direkt nach der einzelnen ParaDesID festgelegt werden können. Das Sparen der obengenannten Attribute kann auch mithilfe des Operators „Select Attributes“ realisiert werden. Das genaue Verfahren ist identisch zur Bereinigung der Redundanzattribute. In Anhang 2 wird ein Screenshot der Datentabelle „ParaDesID“ gezeigt.

Aggregation der bestehenden Attribute unter Zuhilfenahme von Kontextwissen

In Abschnitt 2.3.3 wurde bereits die Theorie über die Themen „Attribut-Extraktion“ und „Attribut-Konstruktion“ vorgestellt. Durch die Datenanalyse wird herausgefunden, dass manche Attribute innere Zusammenhänge miteinander haben und manche Attribute zur besseren Extraktion von nützlichen Kenntnissen kombiniert werden sollen. Damit werden neue Attribute in diesem Experimentabschnitt erstellt und für manche originalen Attribute wird für den späteren DM-Prozess eine Konstruktion durchgeführt.

i. Attribut-Konstruktion (Innere Zusammenhänge zwischen den Attributen)

In diesem Fall wird ein neues Attribut nach dem Kontextwissen erstellt. In der Abbildung 3.8 wird ein Screenshot der Datentabelle „OperationProtocol“ angezeigt. Um die inneren Zusammenhänge besser zu erkennen, sind nur die relevanten Attribute der Datentabelle im Screenshot abgebildet.

WorkpieceGuid	WorkSequence	RoutingSequence	WorkplaceId	ProcessId	Remarks
{728A85D1-9175-4798-B087-00003FB81B1A}	1	10	80	99	AESStart: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	2	20	81	1	Safety check: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	3	30	82	1	Manual test: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	4	40	91	1	HV+EC test: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	5	45	92	1	Switch ON NTM: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	6	50	93	1	Calibration: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	7	60	102	1	Flashing 1: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	8	70	103	1	Flashing 2: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	9	80	113	1	Marking: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	10	85	120	1	DMC-Check: PASS
{728A85D1-9175-4798-B087-00003FB81B1A}	11	90	121	1	Robot: PASS
{93036358-423D-4B57-A573-0000616CB356}	1	10	80	99	AESStart: PASS
{93036358-423D-4B57-A573-0000616CB356}	2	20	81	1	Safety check: PASS
{93036358-423D-4B57-A573-0000616CB356}	3	30	82	1	Manual test: PASS
{93036358-423D-4B57-A573-0000616CB356}	4	40	91	1	HV+EC test: PASS
{93036358-423D-4B57-A573-0000616CB356}	5	45	92	1	Switch ON NTM: PASS
{93036358-423D-4B57-A573-0000616CB356}	6	50	93	1	Calibration: PASS
{93036358-423D-4B57-A573-0000616CB356}	7	60	102	1	Flashing 1: PASS
{93036358-423D-4B57-A573-0000616CB356}	8	70	103	1	Flashing 2: PASS
{93036358-423D-4B57-A573-0000616CB356}	9	80	113	1	Marking: PASS
{93036358-423D-4B57-A573-0000616CB356}	10	85	120	1	DMC-Check: PASS
{93036358-423D-4B57-A573-0000616CB356}	11	90	121	1	Robot: PASS

Abbildung 3.8: Innere Zusammenhänge zwischen Attributen der Datentabelle „OperationProtocol“ (nach Excel)

Durch die Datenanalyse ist es einfach herauszufinden, dass die oben gezeigten fünf Attribute innere Zusammenhänge miteinander haben. Das heißt, dass jedes „WorkpieceGuid“ nach 11 „WorkSequence“ spezialisiert wird und jede „WorkSequence“ einer individuellen „RoutingSequence“, „WorkplaceId“ und „Remarks“ entspricht. Die erste „WorkSequence“ entspricht „Process99“. Deshalb werden in dieser Masterarbeit die Attribute „WorkSequence“, „RoutingSequence“, „WorkplaceId“, „ProcessId“ und „Remarks“ zu einem Attribut kombiniert, damit die besonderen Fälle, die der im Screenshot gezeigten Kombination der Attribute nicht entsprechen, besonders analysiert werden können.

In RapidMiner wird die Aggregation der Attribute mithilfe des Operators „Generate Concatenation“ realisiert. In der Abbildung 3.9 wird ein Screenshot dieses Experimentprozesses gezeigt.

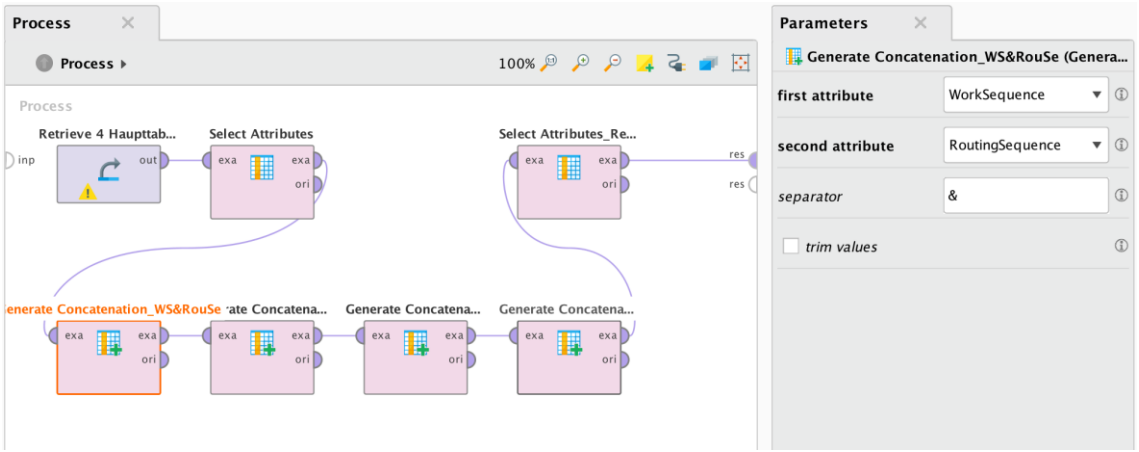


Abbildung 3.9: Modellprozess bei der Kombination der Attribute (nach RapidMiner)

Nach der Einstellung der Inputdaten werden die Attribute „WorkSequence“, „RoutingSequence“, „WorkplaceId“, „ProcessId“, „Remarks“ mithilfe des Operators „Select Attributes“ ausgewählt. Das Auswahlverfahren ist identisch zu dem Verfahren im ersten Abschnitt. Die rechte Seite des Screenshots zeigt das Parameterfenster vom Operator „Generate Concatenation“. Mit dem Parameter „first attribute“ soll das anfängliche Attribut festgelegt werden. Dann

soll das Attribut, mit dem es aggregiert werden soll, als Parameter „second attribute“ festgelegt werden. Mit dem Parameter „separator“ wird das Trennsymbol zwischen zwei Attributen festgelegt werden. Weil für jeden Operator nur zwei Attribute einmal aggregiert werden können, werden insgesamt vier solcher Operatoren eingesetzt. Während der Aggregation werden einige Zwischenkombinationen erstellt, die aber nicht gewünschte Aggregation ist, wie beispielsweise „WorkSequence&RoutingSequence“. Solche Zwischenaggregation werden als Redundanzattribute betrachtet und können mithilfe des Operators „Select Attributes“ bereinigt werden. Der Screenshot der genauen Prozessschritte wird in Anhang 6 angezeigt. In der Abbildung 3.10 zeigt die Statistik der Ergebnisse von diesem Experimentprozess.

Index	Nominal value	Absolute ...	Fraction
1	WS 2&RouSe20&WP81&Process1&Safety check: PASS	18965	0.192
2	WS 4&RouSe40&WP91&Process1&HV+EC test: PASS	16523	0.167
3	WS 3&RouSe30&WP82&Process1&Manual test: PASS	14528	0.147
4	WS 6&RouSe50&WP93&Process1&Calibration: PASS	13180	0.133
5	WS 8&RouSe70&WP103&Process1&Flashing 2: PASS	10996	0.111
6	WS 7&RouSe60&WP102&Process1&Flashing 1: PASS	10176	0.103
7	WS 1&RouSe10&WP80&Process99&AESStart: PASS	4438	0.045
8	WS 9&RouSe80&WP113&Process1&Marking: PASS	1024	0.010

Abbildung 3.10: Statistik der aggregierten Attribute von der HDT (nach RapidMiner)

Durch Statistik in der Abbildung 3.10 ist deutlich zu erkennen, dass die kombinierten Attribute, die den inneren Zusammenhängen entsprechen, den höchsten prozentualen Anteil haben. Das heißt, dass die diese fünf Attribute innere Zusammenhänge haben.

ii. Attribut-Extraktion

In diesem Abschnitt wird eine Konstruktion für zwei originale Attribute unter Berücksichtigung des Bedarfs des DM-Prozesses durchgeführt. Nach der Datenanalyse wird herausgefunden, dass das Datenformat der Attribute „BeginOfManufacturing“ und „EndOfManufacturing“ komplex ist und diese Attribute zur Reduzierung der Komplexität aggregiert werden sollen. Das Format der beiden Attribute enthält die Zeiteinheiten „Jahr“, „Monat“, „Tag“, „Stunde“, „Minute“, „Sekunde“ und „Millisekunde“. Ein solches komplexes Format ist schwer im späteren DM-Prozess zu analysieren. Weiterhin entsprechen diese beiden Attribute nicht der Regel der Atomarität. Das heißt, dass noch die Möglichkeit besteht, die beiden Attribute weiter zu trennen. Unter Berücksichtigung von Kontextwissen kam die Idee auf, den Zeitabstand zwischen den beiden Attributen zu berechnen. Die Aggregation der beiden Attribute bringt viele Vorteile. Zuerst wird mithilfe der Aggregation ein Attribut gespart und der Rechenaufwand für den späteren DM-Prozess wird somit reduziert. Zweitens enthält das Datenformat des neuen Attributes nicht mehr die obengenannte 7 Zeiteinheiten, sondern nur eine Zeiteinheit zur Repräsentation des Zeitabstands. Drittens ist das neue Attribut nicht trennbar und entspricht den Regel der

Atomarität. Viertens vereinfacht das die Datenanalyse im späteren DM-Prozess durch die Reduzierung der Komplexität.
Die Abbildung 3.11 zeigt einen Screenshot dieses Experimentprozesses.

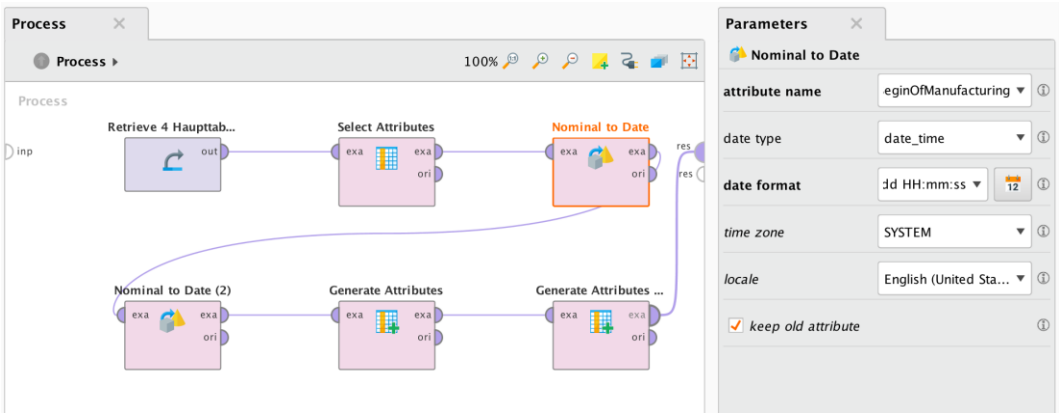


Abbildung 3.11: Modellprozess der Aggregation der Attribute „BeginOfManufacturing“ und „EndOfManufacturing“ (nach RapidMiner)

Identisch zu letzten Experimentprozessen wird zuerst der Operator „Retrieve“ eingesetzt für die Datenbereitstellung. Danach wird der Operator „Select Attributes“ eingesetzt, um die relevanten Attribute „BeginOfManufacturing“ und „EndOfManufacturing“ auszuwählen. Nach der Vorbereitung beginnt der Datentyptransformations-Prozess. Beim anfänglichen Datenimport werden die beiden relevanten Attribute als Datentyp „polynominal“ automatisch erkannt. Aber die Funktion zur Berechnung des Zeitabstands gilt nur für den Datentyp „date-type“. Deshalb muss der Datentyp der beiden Attribute von „polynominal“ zu „date time“ transformiert werden. Auf der rechten Seite des Screenshots wird die genaue Parametereinstellung des Operators „Nominal to Date“ angezeigt. Der Parameter *attribute name* wird mit dem Attribut „BeginOfManufacturing“ festgelegt. Der Parameter *date type* wird mit der Option *date_time* festgelegt, damit die transformierte Zeiteinheit sowohl das Datumformat als auch das Zeitpunktformat hat. Der nächsten Parameter *date format* wird mit dem Format „yyyy-MM-dd HH:mm:ss“ festgelegt. Datentyptransformation gehört zum Thema „*Datenhomogenisierung*“ und wird in Abschnitt 3.2.2 genau behandelt. Nach der Datentyptransformation wird der Zeitabstand mithilfe des Operators „Generate Attributes“ berechnet. Gleichzeitig wird ein neues Attribut „ManufacturingTime(Second)“ zur Repräsentation des Zeitabstands generiert. Im Operator „Generate Attributes“ wird das neue Attribut mithilfe einer Funktion *„date_diff (Datefirst, Datesecond)“* realisiert. Zum Schluss werden das Zwischenredundanzattribut „ManufacturingTime“ und die beiden originalen Attribute „BeginOfManufacturing“ und „EndOfManufacturing“ mithilfe des Operators „Select Attributes“ bereinigt. Nun wird eine Abbildung vom Ergebnis dieses Experimentprozesses gezeigt. Um den Experimentprozess besser zu verstehen, sind die originalen zwei Attribute stehen geblieben.

BeginOfManufacturing	EndOfManufacturing	Manufactur...
May 18, 2015 9:02:07 PM CEST	May 18, 2015 9:17:36 PM ...	929
May 18, 2015 9:02:07 PM CEST	May 18, 2015 9:17:36 PM ...	929
May 18, 2015 9:02:07 PM CEST	May 18, 2015 9:17:36 PM ...	929

Abbildung 3.12: Modellierungsergebnis der Aggregation der Attribute „BeginOfManufacturing“ und „EndOfManufacturing“ (nach RapidMiner)

Die Abbildung 3.12 zeigt den Prozess durch die tatsächlichen Daten, damit der Experimentprozess besser verstanden kann. In Anhang 7 werden die genauen Schritte dieses Modellprozesses mithilfe von Screenshots gezeigt.

Diskretisierung

Die letzten drei genannten Datenaggregationsverfahren konzentrierten sich auf die Aggregation von Attributen, während die Diskretisierung sich um die Aggregation der Attributwerte eines Attributes kümmert. Durch die Datenanalyse wird herausgefunden, dass viele Attribute, die eine hohe Summe an Datensätzen haben, nach neuen diskretisierten Attributwerten aggregiert werden können. Damit können alle Datensätze eines Attributes neuen diskretisierten Attributwerten zugeordnet werden und der Rechenaufwand der diskretisierten Attribute kann im späteren DM-Prozess reduziert werden, weil die Summe der neuen Attributwerte reduziert ist.

In RapipMiner wird die Diskretisierung mithilfe des Operators „Discretize“ realisiert. Nach der Datenanalyse wird entschieden, zwei Attribute „NmbOfRepairs“ und das neu generierte Attribut „ManufacturingTime(Second)“ zu diskretisieren. Die Abbildung 3.13 zeigt den Experimentprozess von der Diskretisierung.

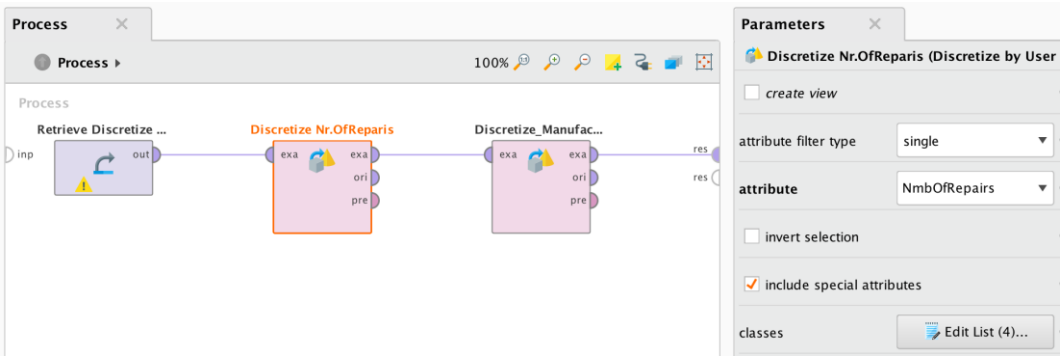


Abbildung 3.13: Modellprozess der Diskretisierung der Attribute „NmbOfRepairs“ und „ManufacturingTime(Second)“ (nach RapidMiner)

Identisch zu den letzten Experimenten werden die Daten der beiden Attribute mithilfe des Operators „Retrieve“ als Dateninput eingesetzt. Danach werden beide Attribute hintereinander mithilfe des Operators „Discretize by User Specification“ diskretisiert. Durch die Datenanalyse der Top 100.000 HDT wird herausgefunden, dass die maximale Reparierhäufigkeit 7 beträgt. Deshalb werden vier neue unterschiedlich diskretisierte Attributwerte nach der Anzahl der Reparierhäufigkeit generiert, nämlich „no repair“, „low repair“, „middle repair“, „high repair“. Das Attribut „ManufacturingTime(Second)“ ist etwas komplexer. Laut dem Ergebnis des letzten Modelles, der Aggregation der Attribute „BeginOfManufacturing“ und „EndOfManufacturing“,

ist die Länge der Produktionszeit in der Zeiteinheit „Sekunde“ zu hoch und die Standardabweichung dieses Attributes ist entsprechend auch hoch. Deshalb werden für die neuen diskretisierten Attributwerte von der Diskretisierung die Zeiteinheiten „Minute“ und „Stunde“ angewendet. Nach einigen Tests werden letztendlich 7 unterschiedlich diskretisierte Attributwerte nach der Länge der Produktionszeit generiert, nämlich „<5 min“, „5-10 min“, „10-15 min“, „15-30 min“, „30-60 min“, „1-2 h“ und „>2 h“. Die Abbildung 3.14 vergleicht die Statistik der beiden Attribute vor und nach der Diskretisierung.

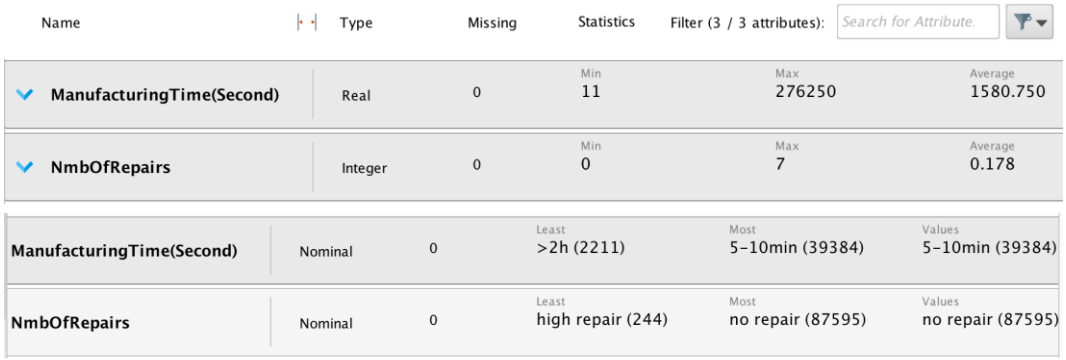


Abbildung 3.14: Vergleich der Modellierungsergebnisse vor und nach dem Diskretisierungsprozess von den Attributen „NmbOfRepairs“ und „ManufacturingTime(Second)“ (nach RapMin)

Mithilfe der dargestellten Statistik ist zu erkennen, dass die Summe der Attributwerte von beiden Attributen durch den Diskretisierungsprozess stark reduziert wird. Damit wird auch der Rechenaufwand für den späteren DM-Prozess stark reduziert und die Visualisierung der Experimentergebnisse wird auch vereinfacht. In Anhang 8 werden die Parametereinstellung und das Experimentergebnis gezeigt.

Neben der Anwendung auf die Datenaggregation wird die Methode „Diskretisierung“ auch bei der Datentyptransformation und der Bereinigung der Ausreißer angewendet. Die beiden Bereiche werden im nächsten Abschnitt „Datenhomogenisierung“ weiter behandelt.

3.2.2 Datenhomogenisierung

Nach der vorherigen Datenanalyse ist deutlich geworden, dass die Datenqualität der originalen Daten nicht gut ist. In manchen Attributen gibt es zahlreiche fehlende und verrauschte Daten. Der Datentyp von allen Attributen ist nicht gleich. Weiterhin wird der Datentyp von den Attributen „TotalResult“ und „Status“ von RapidMiner als „binominal“ erkannt, weil die beiden Attribute nur zwei unterschiedliche Attributwerte haben und die fehlende Werte der beiden Attribute nicht direkt mithilfe des normalen Verfahrens bereinigt werden können. Deshalb ist es notwendig, die originalen Daten zu homogenisieren.

Nach dem Bedarf der Datenanalyse werden folgende vier Aufgaben der Datenhomogenisierung im diesem Abschnitt erläutert:

1. Prüfung auf Format-Standardisierung von den Datenbeständen und auf Atomarität der Attribute
2. Bereinigung der fehlenden Werte
3. Bereinigung der verrauschten Daten
4. Datentyptransformation

Im Folgenden wird jede Aufgabe entsprechend dem Experimentprozess genau erläutert.

Prüfung auf Format-Standardisierung von den Datenbeständen und auf Atomarität der Attribute

Bevor die Datenvorverarbeitung anfängt, werden die Inputdaten nach dem Vorgehensmodell zur MESC von [ITP16] auf zwei wichtige Kriterien geprüft, nämlich die Format-Standardisierung und die Atomarität. Mithilfe dieser beiden Kriterien wird entschieden, ob die Inputdaten für die Datenanalyse geeignet sind.

Jeder Datenbestand umfasst viele Datentabellen und jede Tabelle enthält redundante Daten, aber nicht jede Datentabelle soll im späteren DM-Prozess analysiert werden. Deshalb werden vier HDT ausgewählt und zu einer gemeinsamen Datentabelle gruppiert. Damit wird der originale Datenbestand in ein Standardformat überführt.

Das zweite Prüfungskriterium heißt „Attribut-Atomarität". Es ist besonders wichtig, die Atomarität der Attribute sicherzustellen, die später analysiert werden sollen. Die Atomarität eines Attributes bedeutet, dass das Attribut nicht weiter aufgeteilt werden kann [FR06, S. 5]. Es folgt ein Beispiel: Ein Attribut einer Datentabelle heißt Kontaktinformation und dieses Attribut kann in zwei weitere Attribute aufgeteilt werden, nämlich Handynummer und Festnetznummer. In diesem Fall wird gegen die Regel der Atomarität verstoßen. Nach der Datenanalyse wird es herausgefunden, dass fast alle Attribute von den vier HDT nicht weiter aufgeteilt werden können Die Ausnahmen sind solche Attribute wie „BeginOfManufacturing" und „EndOfManufacturing". Die beiden Attribute enthalten viele Zeiteinheiten und können theoretisch noch weiter aufgeteilt werden. Nach der Aggregation der beiden Attribute im letzten Abschnitt hat das neue Attribut nur noch eine Zeiteinheit und kann nicht weiter aufgeteilt werden.

Bereinigung der fehlenden Werte

Nach der manuellen Reduktion der nutzlosen Attribute im letzten Abschnitt „Datenaggregation" ist die Summe der Attribute stark reduziert, aber viele übrige Attribute enthalten fehlende Werte bei den Datensätzen, die nicht geeignet für den späteren DM-Prozess sind. In Abschnitt 2.3.2 wurden einige Methoden zur Bereinigung der fehlenden Werte schon vorgestellt. Durch die Datenanalyse wird entschieden, die fehlenden Werte direkt zu bereinigen. Die Gründe dafür sind: Der Prozentsatz der fehlenden Werte innerhalb der übrigen Attribute ist klein. Deshalb hat eine direkte Bereinigung der fehlenden Werte fast keinen Einfluss auf die Ergebnisse des späteren DM-Prozesses. In Anhang 9 wird eine Tabelle über die Summe der fehlenden Werte von den Attributen gezeigt. Die Summe der Stichprobe beträgt 100.000.

Die Bereinigung der fehlenden Werte von den Datensätzen wird in zwei Schritten durchgeführt. Zuerst werden alle fehlenden Werte durch die Methode „*Technische Kodierung*" mithilfe einer besonderen Nummer ersetzt, um die originalen Daten nicht zu verfälschen und später einheitlich filtern zu können. Danach werden die kodierten fehlenden Werte gefiltert. Damit werden alle fehlenden Werte bereinigt.

Ersetzung der fehlenden Werte

Die Ersetzung der fehlenden Werte wird durch den Operator „Replace Missing Values" realisiert. Aus technischen Gründen können in RapidMiner die Daten nur mit einer bestimmten Ziffer ersetzt werden. Um die originalen Daten nicht zu verfälschen, werden die fehlenden Werte mit einer Ziffer „9999999999" ersetzt. In der Abbildung 3.15 wird der Modellprozess der Ersetzung der fehlenden Werte gezeigt.

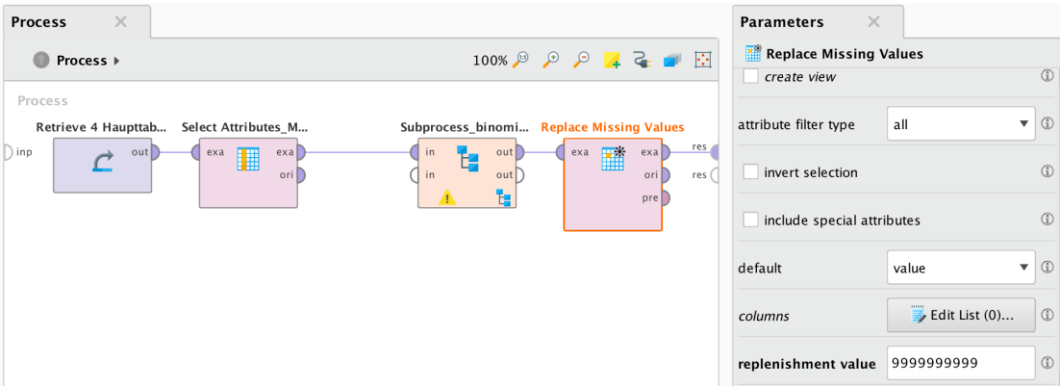


Abbildung 3.15: Modellprozess der Ersetzung der fehlenden Werte (nach RapidMiner)

Nach den Vorbereitungsschritten wird der Datentyp von den Attributen „TotalResult“ und „Status“ von „binominal“ zum „polynominal“ transformiert. Der Transformationsprozess wird später im letzten Abschnitt „Datentyptransformation“ genau erläutert. Zum Schluss werden die fehlenden Werte mithilfe des Operators „Replace Missing Values“ durch die Nummer „999999999“ ersetzt. Für den Parameter *attribute filter type* ist die Option *all* auszuwählen, weil alle fehlenden Werte durch eine einheitliche Nummer ersetzt werden sollen. Für den Parameter *default* wird die Option *value* ausgewählt, weil die fehlenden Werte mit einer konkreten Nummer ersetzt werden. Mit dem letzten Parameter wird die ersetzende Ziffer festgelegt.

Filterung der fehlenden Werte

Nach der Kodierung der fehlenden Werte sollen solche Werte wieder gefiltert werden, um die fehlenden Werte zu bereinigen. Die Filterung wird mithilfe des Operators „Filter Examples“ realisiert. In der Abbildung 3.16 wird der Experimentprozess mithilfe eines Screenshots vorgestellt.

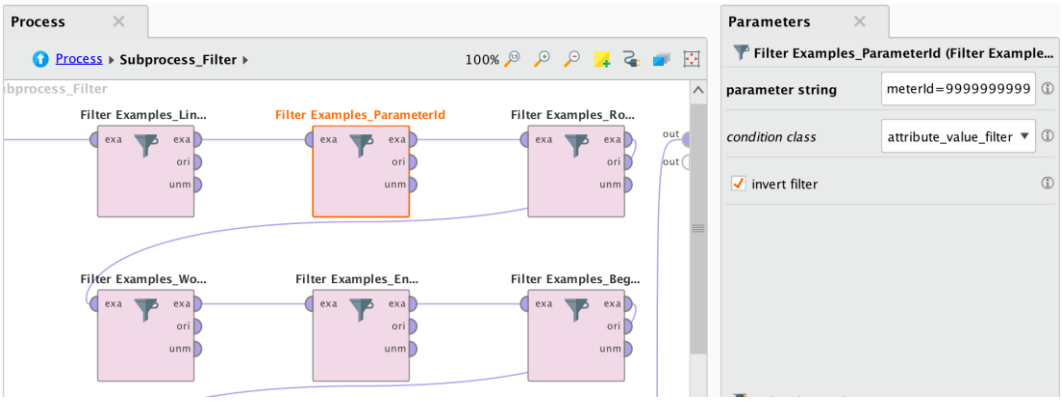


Abbildung 3.16: Modellprozess der Filterung der fehlenden Werte (nach RapidMiner)

Im Experimentprozess werden die kodierten Werte aller relevanten Attribute nacheinander mithilfe des Operators „Filter Examples“ gefiltert. Für den Parameter *condition class* ist die Option *attribute_value_filter* auszuwählen, weil eine bestimmte Nummer gefiltert werden sollen. Beim Parameter *parameter string* soll eine Formel mit dem Format „Attribut=Nummer“ eingegeben werden. Es ist besonders darauf zu achten, dass die Option *invert filter* gewählt ist, weil die kodierten Werte gefiltert, aber nicht ausgewählt werden sollen.

Bereinigung der verrauschten Daten

Die verrauschten Daten sind zwar reale Daten, aber sie können das Ergebnis des späteren DM-Prozesses verfälschen. Deshalb sollen die verrauschten Daten bereinigt werden. In dieser Masterarbeit werden zwei Verfahren zur Bereinigung der verrauschten Daten angewendet, nämlich die Diskretisierung und die direkte Bereinigung der verrauschten Daten.

Diskretisierung

Im letzten Abschnitt „Datenaggregation“ wurde die Methode „Diskretisierung“ zur Reduktion der Attributwerte eines Attributes angewendet. Bei der Bereinigung der verrauschten Daten kann diese Methode auch eingesetzt werden. Aber diese Methode ist nur geeignet für die Attribute, die nicht viele unterschiedliche Attributwerte haben. Die Funktionsweise ist: Die nützlichen Datensätze eines Attributs werden entsprechenden Attributwerten zugeordnet und nach solchen Attributwerten diskretisiert. Die verrauschten Daten gehören nicht zu solchen Attributwerten, die nach dem Bedarf manuell festgelegt werden, damit die verrauschten Daten bereinigt werden. Nach Datenanalyse wird entschieden, die verrauschten Daten von den Attributen „LineId“, „LastRoutingSequence“, „NextRoutingSequence“, „WorkSequence“, „RoutingSequence“, „WorkplaceId“, „ResultSequence“ und „ProcessId“ mithilfe dieser Methode zu bereinigen.

Identisch zu letzten Abschnitt „Datenaggregation“ wird die Diskretisierung auch mithilfe des Operators „Discretize“ realisiert. Der Experimentprozess ist identisch zum Diskretisierungsprozess im letzten Abschnitt und wird hier nicht nochmals beschrieben. Als ein Beispiel wird die Parametereinstellung des Attributs „LineId“ in der Abbildung 3.17 angezeigt.

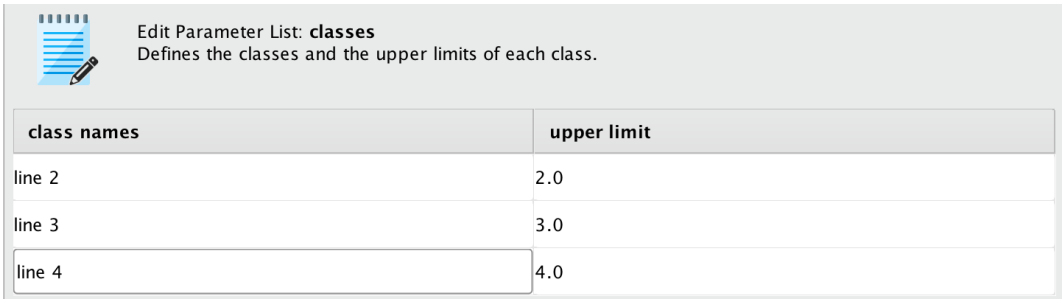


Abbildung 3.17: Diskretisierung des Attributs „LineId“ zur Bereinigung der verrauschten Daten (nach RapidMiner)

Durch die Überprüfung der Daten in der Software SQL wird herausgefunden, dass das Attribut „LineId“ drei unterschiedliche Attributwerte „2“, „3“ oder „4“ hat. Deshalb werden auch drei entsprechende „class names“ generiert, nämlich „line 2“, „line 3“ und „line 4“. Damit werden die verrauschten Daten bereinigt, die den obengenannten drei originalen Attributwerten nicht entsprechen. In Anhang 10 wird die Parametereinstellung von anderen Attributen angezeigt.

Direkte Bereinigung der verrauschten Daten

Manche deutlich verrauschten Datenwerte können durch die Methode „technische Kodierung“ und die anschließende Filterung direkt bereinigt werden, z. B. „NONE“, „NULL“, „N“. Der Experimentprozess ist fast identisch zu letzter Experimentprozess „Ersetzung der fehlenden Werte“. Der angewendete Operator ist in diesem Fall nicht „Replace Missing Values“, sondern „Replace“. Die genaue Funktionsweise ist: Die obengenannten verrauschten Datenwerte werden

zuerst mithilfe des Operators „Replace“ mit der Nummer „9999999999“ kodiert und die kodierten Daten werden anschließend mithilfe des Operators „Filter Examples“ gefiltert. Der Experimentprozess der Filterung der kodierten verrauschten Daten ist identisch zum Prozess vom letzten Abschnitt „Filterung der fehlenden Werte“ und wird nicht nochmals beschrieben. Der Modellprozess wird in der Abbildung 3.18 gezeigt.

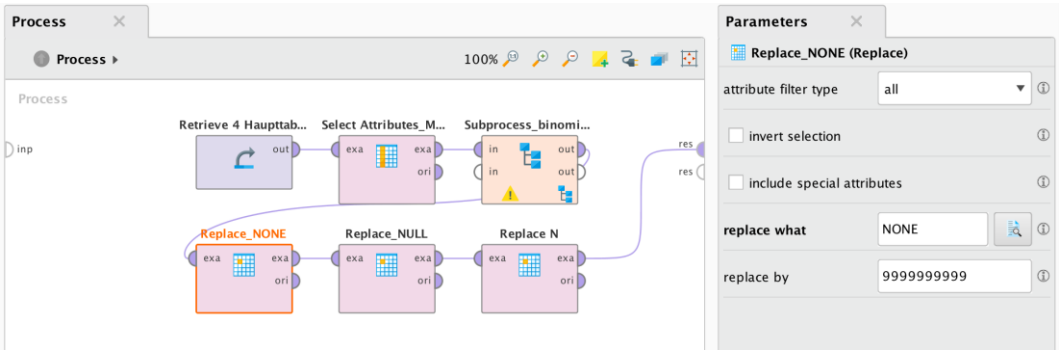


Abbildung 3.18:Modellprozess der Ersetzung der verrauschten Daten (nach RapidMiner)

Nach dem Vorbereitungsprozess werden drei Operatoren „Replace“ eingesetzt, um die verrauschten Datenkategorien „NONE“, „NULL“ und „N“ durch die Nummer „9999999999“ zu ersetzen. Damit werden diese drei verrauschten Datenwerte kodiert und können später mithilfe des Operators „Filter Examples“ direkt gefiltert werden. Die rechte Seite des Screenshots zeigt die Parametereinstellung des Operators „Replace_NONE“. Für den Parameter *attribute filter type* wird die Option *all* gewählt, weil die Datenkategorie „NONE“ bei allen Attributen kodiert werden soll.

Datentyptransformation

Durch die Datenanalyse wird herausgefunden, dass die meisten numerischen Daten jedoch keine numerische Bedeutung haben, z. B. das Attribut „Tag“. Die Datenwerte sind zwar Zahlen, aber haben keine numerische Bedeutung, weil das Etikett nur eine Vertretung jedes Arbeitslaufs ist und individuell zu jedem Ablauf gehört. Deshalb sollen solche „unechten“ numerischen Daten zu nominalen Daten transformiert werden. Um die fehlenden Werte der Attribute „TotalResult“ und „Status“ nach den normalen Verfahren bereinigen zu können, muss der Datentyp der beiden Attribute von „binominal“ zu „polynominal“ transformiert werden. Um den Zeitabstand zwischen dem Attribut „BeginOfManufacturing“ und „EndOfManufacturing“ berechnen zu können, soll der Datentyp dieser beiden Attribute von „polynominal“ zu „date-time“ transformiert werden. Die Datentyptransformation kann entweder mithilfe eines Datentyptransformations-Operators, z. B. „Nominal to Numerical“, oder mithilfe der Methode „Diskretisierung“ realisiert werden. In diesem Abschnitt werden die beiden Verfahren sowie die Transformation des Datentyps „binominal“ nach dem jeweiligen Modellprozess genau erläutert.

Transformation des Datentyps „binominal“

Der Datentyp der Attribute „TotalResult“ und „Status“ wird beim Datenimport automatisch von RapMin als Datentyp „binominal“ erkannt, weil die beiden Attribute nur zwei unterschiedliche Attributwerte haben. Deshalb kann der Modellprozess der Bereinigung der fehlenden Werte nicht funktionieren bei den beiden Attributen, weil die Nummer „9999999999“ als dritter Attri-

butwert hinzugefügt werden soll. Damit ist es notwendig, den Datentyp von „binominal“ zu „polynominal“ zu transformieren.

In RapidMiner wird dieser Modellprozess mithilfe der Operatoren „Nominal to Numerical“ und „Discretize“ realisiert. Die Funktionsweise ist: Zuerst wird der Datentyp von „binominal“ zu „numerical“ mithilfe des Operator „Nominal to Numerical“ transformiert. Danach wird der Datentyp mithilfe des Operators „Discretize“ zu „nominal“ transformiert. In der Abbildung 3.19 wird der Experimentprozess mithilfe eines Screenshots gezeigt.

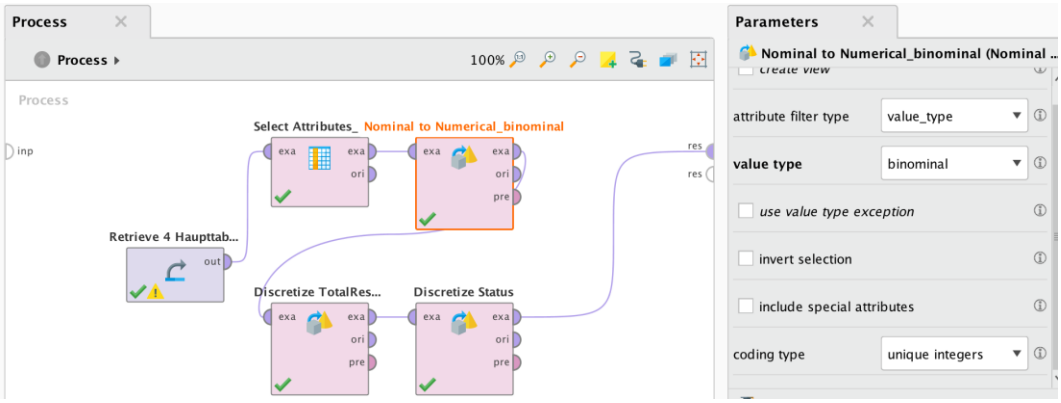


Abbildung 3.19: Modellprozess der Transformation des Datentyps „binominal“ (nach RapMiner)

Nach den ersten zwei Vorbereitungsschritten wird der Datentyp der beiden Attribute von „binominal“ zu „numerical“ mithilfe des Operators „Nominal to Numerical“ transformiert. Die rechte Seite des Screenshots zeigt die Parametereinstellung dieses Operators. Der Parameter „attribute filter type“ wird mit der Option „value_type“ festgelegt, damit im nächsten Schritt für den Parameter „value_type“ der Datentyp „binominal“ festgelegt werden kann. Für den Parameter „coding type“ wird die Option „unique integers“ festgelegt, damit die zwei Attributwerte der beiden Attribute mit den Werten „0“ und „1“ kodiert werden können. Nach der Kodierung wird der Datentyp von „numerical“ zu „polynominal“ mithilfe des Operators „Discretize“ transformiert. In Anhang 11 wird ein Screenshot der Parametereinstellung des Operators „Discretize“ angezeigt.

Datentyptransformation durch die Datentyp-Operatoren

Gemäß der Erklärung am Anfang dieses Abschnitts sollen die unechten numerischen Daten zum Datentyp „nominal“ transformiert werden, um das Ergebnis des späteren DM-Prozesses nicht zu verfälschen. Es stehen einige nützliche Operatoren in RapidMiner für die direkte Datentyptransformation zur Verfügung, z. B. „Numerical to Polynominal“, „Numerical to Date“, „Nominal to Numerical“ usw. Im letzten Abschnitt „Datenaggregation“ wird der Operator „Numerical to Date“ auf die Berechnung des Zeitabstands zwischen den Attributen „BeginOfManufacturing“ und „EndOfManufacturing“ angewendet. Der Operator „Nominal to Numerical“ wurde gerade im letzten Abschnitt zur Behandlung des Datentyps „binominal“ angewendet. In diesem Abschnitt wird nun der Operator „Numerical to Polynominal“ zur Transformation des Datentyps von „numerical“ zu „polynominal“ eingesetzt. Dieser Operator ist geeignet für solche Attribute, die zahlreiche unterschiedliche Attributwerte umfassen, weil der Datentyp aller Attributwerte des Attributs einmal zusammen transformiert wird und keine neuen Attributwerte generiert werden sollen. Durch die Datenanalyse wird der Datentyp der Attribute „Tag“ und „Se-

rialNumber1“ mithilfe dieses Operators transformiert. In der Abbildung 3.20 wird der Screenshot des Modellprozesses gezeigt.

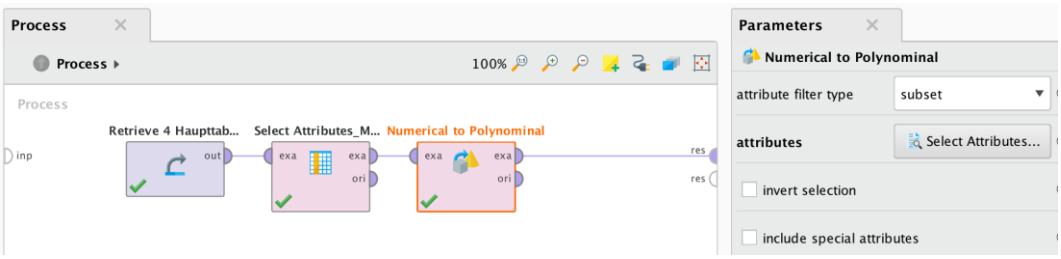


Abbildung 3.20: Modellprozess der direkten Transformation des Datentyps (nach RapidMiner)

Die Vorbereitungsschritte sind identisch zu den letzten Experimentprozessen. Der Parameter „attribute filter type“ soll mit der Option „subset“ ausgewählt werden, damit die zwei Attribute „Tag“ und „SerialNumber1“ ausgewählt werden können.

Diskretisierung

Wenn die Summe an Attributwerten eines Attributes nicht groß ist, funktioniert die Methode „Diskretisierung“ zur Datentyptransformation besser, weil jeder Attributwert nach dem Bedarf neu benannt werden kann und einige Attributwerte nach Bedarf zur Reduktion der Summe der Attributwerte sowie der Komplexität des Datasets aggregiert werden können. Ähnlich wurde dies schon im letzten Abschnitt „Datenaggregation“ erläutert. Weiterhin kann die Methode „Diskretisierung“ zur Bereinigung der verrauschten Daten angewendet werden und auch diese Anwendung wurde schon im letzten Abschnitt „Bereinigung der verrauschten Daten“ genau erläutert. Nach der Datenanalyse wird entschieden, den Datentyp der Attribute „LineId“, „LastRoutingSequence“, „NextRoutingSequence“, „WorkSequence“, „RoutingSequence“, „WorkplaceId“, „ResultSequence“ und „ProcessId“ mithilfe der Methode „Diskretisierung“ zu transformieren. Der Modellprozess dieses Abschnitts ist identisch zum Prozess im Abschnitt „Bereinigung der verrauschten Daten“. Das heißt: Durch die Durchführung des Diskretisierungsprozesses von den obengenannten Attributen werden sowohl der Datentyp von „numerical“ zu „polynominal“ transformiert als auch die verrauschten Datenwerte bereinigt. In Anhang 10 wird die Parametereinstellung von allen diskretisierten Attributen angezeigt.

3.2.3 Feature Selection

In dieser Masterarbeit werden zwei Feature Selection (FS)-Methoden im Modellprozess angewendet, nämlich manuelle Auswahl von Attributen und Chi-Square-Statistik. Die Durchführung der ersten Methode wurde in Abschnitt 3.2.1 genau behandelt und somit hier nicht wiederholt. Dieser Abschnitt konzentriert sich auf die zweite Methode „Chi-Square-Statistik“.

Die Funktionsweise der Chi-Square-Statistik im Modellprozess ist die Berechnung der Attributrelevanz bezüglich des Label-Attributs. Je größer das Ergebnis ist, desto wichtiger ist das Attribut für das Label-Attribut. In RapieMiner wird der Modellprozess mithilfe von drei Attributen „Set Role“, „Weight by Chi Squared Statistic“ und „Select by Weights“ realisiert. Der Experimentprozess FS soll nach der Datenhomogenisierung und einigen Schritten zur Datenaggregation durchgeführt werden. Eine homogene Datentabelle ist eine wichtige Voraussetzung für FS, weil eine inhomogene Datentabelle das Endergebnis verfälschen kann. Mithilfe der Datenaggregation kann der Rechenaufwand der Methode „Chi Squared Statistic“ reduziert werden, weil

nach der Datenaggregation die Summe der Attributwerte des Attributs reduziert wird. Um den Experimentprozess nicht zu komplex zu machen, wird in der Abbildung 3.21 nur der Unterprozess der Methode „Chi Squared Statistic“ durch einen Screenshot gezeigt.

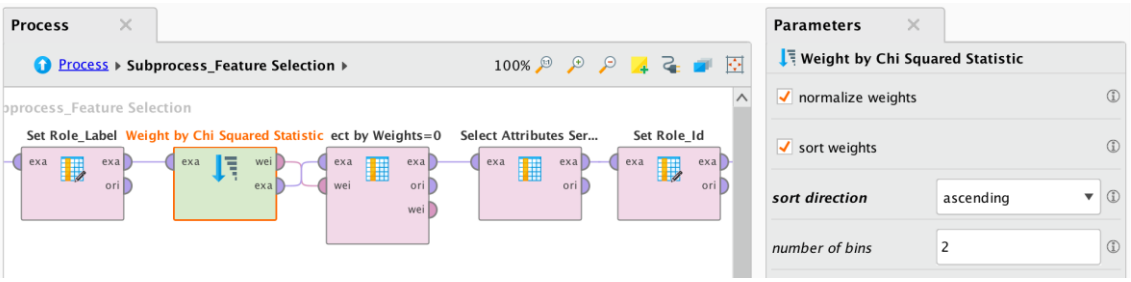


Abbildung 3.21: Modellprozess der FS-Methode „Chi Square-Statistik“ (nach RapidMiner)

Zuerst wird ein Label-Attribut mithilfe des Operators „Set Role“ festgelegt. Nach dem Ergebnis des Datenauswahlprozesses im Abschnitt 3.2.1 wird das Attribut „Tag“ als das Label-Attribut festgelegt. Dann wird die Attributrelevanz aller Attribute auf Grundlage des Label-Attributs mithilfe der Methode „Chi Square Statistik“ berechnet. Weil die Datentabelle 100.000 Zeilen hat, kann der genaue Berechnungsprozess hier nicht dargestellt werden. Das Berechnungsverfahren basiert auf die Kalkulation der Häufigkeit von unterschiedlichen Attributen. Ein einfaches Beispiel wurde schon in Abschnitt 2.3.4 präsentiert. Dieses Beispiel zeigt das Berechnungsverfahren dieser Methode. Wenn das Ergebnis gleich 0 ist, bedeutet das, dass die entsprechenden Attribute identisch zum Label-Attribut sind, also Duplikate. Ein Ergebnis gleich 1 besagt, dass die entsprechenden Attribute gar keine Relevanz für das Label-Attribut haben, also vollständig unabhängig sind. Somit können die Duplikate direkt gefiltert werden. Wenn das Ergebnis gleich 1 ist, soll es nicht direkt gefiltert werden: In diesem Fall bedeutet es, dass die entsprechenden Attribute zwar theoretisch unabhängig vom Label-Attribut sind, aber nach dem praktischen Gesichtspunkt einige Attribute bereinigt werden sollten. Deshalb werden nur die Ergebnisse, die gleich 0 sind, mithilfe des Operators „Select by Weights“ gefiltert. Die Ergebnisse, die gleich 1 sind, werden mithilfe des Operators „Select Attributes“ manuell unter Berücksichtigung der praktischen Nützlichkeit bereinigt. Nach der Aufgabenstellung soll später die Clusteranalyse als DM-Verfahren durchgeführt werden, wobei diese ein nicht überwachtes DM-Verfahren ist. Deshalb ist ein Label-Attribut nicht notwendig für den späteren DM-Prozess. Mithilfe des Operators „Set Role“ wird für das Attribut „Tag“ der Attributtyp „id“ festgelegt. Die rechte Seite des Screenshots zeigt die Parametereinstellung des Operators „Weight by Chi Squared Statistic“. Die Option *normalize weights* soll angekreuzt werden, damit das Ergebnis innerhalb des Intervalls [0, 1] angezeigt wird. Die Option *sort weights* soll auch gewählt werden, damit die Ergebniswerte nach der Reihenfolge gelistet werden. Mithilfe der beiden Optionen wird das Ergebnis einfacher zu lesen. Der Parameter *number of bins* dient zur Behandlung der Daten vom Datentyp „numerical“ und ist nicht relevant für das Ergebnis, weil der Experimentprozess der Diskretisierung vorher schon durchgeführt und der Datentyp aller Daten schon zu „nominal“ transformiert wurde. In Anhang 12 werden die Parametereinstellung und das Zwischenergebnis des Experimentprozesses mithilfe eines Screenshots gezeigt. Das Berechnungsverfahren für den Gewichtungswert durch das Chi-Square-Verfahren ist ähnlich wie beim Beispiel in Abschnitt 2.3.4. Im Vergleich mit dem Beispiel sollen die zwei Attribute beim Expe-

riment „Tag“ und irgendein anderes Attribut sein. Das heißt, dass das Label-Attribut „Tag“ mit einem anderen Attribut zusammen eine Kontingenztabelle bildet. Weil die Attribute von den Firmendaten praktische Bedeutung haben, wird der erhaltene X^2-Wert nicht für die Beurteilung der Ablehnung der Hypothese berücksichtigt.

3.3 Aufbau des vollständigen Modelles

Nach dem Aufbau der einzelnen Experimentprozesse werden all diese gemeinsam zu einem vollständigen Experimentmodell zusammengeführt. Dazu wird in der Abbildung 3.22 ein Screenshot des Modellprozesses gezeigt.

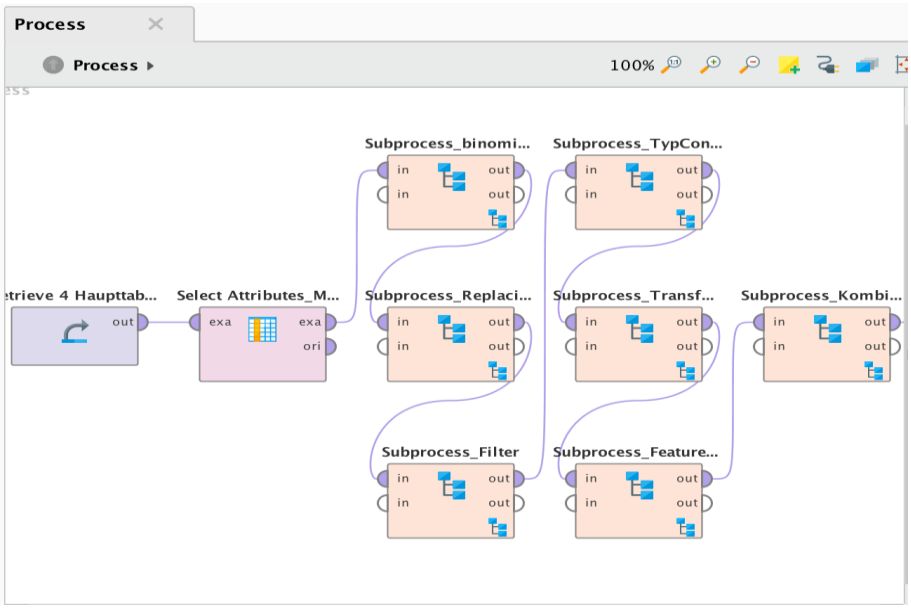


Abbildung 3.22: Vollständiges Modell zur Datenvorverarbeitung (nach RapidMiner)

Am Anfang wird der Operator „Retrieve“ zur Datenbereitstellung eingesetzt. Dann wird der Operator „Select Attributes“ angewendet, um die durch die Datenanalyse festgelegten nützlichen Attribute auszuwählen. Nach der Vorbereitungsphase wird der „Subprocess_binominal“ durchgeführt, um den Datentyp von den relevanten Attributen von „binonimal“ zu „nominal“ zu transformieren. Anschließend wird der Modellprocess „Replace Missing Values“ durchgeführt, damit alle fehlende Werte sowie die deutlich verrauschten Daten „NONE“, „NULL“ und „N“ mit der Nummer „9999999999“ einheitlich kodiert werden. Danach wird der „Subprocess_Filter“ durchgeführt, wobei alle kodierten fehlenden und verrauschten Daten bereinigt werden. Nach der Filterung wird die direkte Transformation des Datentyps von den Attributen „Tag“ und „Status“ durgeführt, die eine hohe Summe an Attributwerten haben, damit der Datentyp von beiden Attributen von „numerical“ zu „nominal“ transformiert wird. Anschließend wird die Datentyptransformation von den Attributen, die eine geringe Summe an Attributwerten haben, mithilfe der Methode „Discretize“ durchgeführt. Gleichzeitig werden auch die verrauschten Daten von den diskretisierten Attributen bereinigt. Nach der Datenhomogenisierung wird der „Subprocess_Feature Selection“ durch die Methode „Chi Square Statistic“ durchgeführt. Die übrigen Attribute nach der manuellen Attributauswahl werden durch die Berechnung der jeweiligen Attributsrelevanz mit dem Label-Attribut „Tag“ nochmals ausgewählt und gefiltert. Der

letzte „Subprocess_Kombination der Attribute“ konzentriert sich auf die Datenvorverarbeitungsmethode „Datenaggregation“. Dieser „Subprocess“ unterteilt sich in zwei Abschnitte, nämlich die Berechnung des Zeitabstands zwischen den Attributen „BeginOfManufacturing“ und „EndOfManufacturing“ sowie die Aggregation der Attribute, die innere Zusammenhänge haben.

3.4 Visualisierung und Interpretation der Ergebnisse

In diesem Abschnitt werden nun die Ergebnisse der Datenvorverarbeitung mithilfe von Tabellen und Balkendiagrammen visualisiert. Nach der Fragestellung soll das Clusterverfahren im späteren DM-Prozess angewendet werden. Ein Clusterverfahren ist ein „unsupervised“ Verfahren und es gibt am Anfang keine deutlichen Klassifikationskriterien oder Schwerpunkte, nach denen der DM-Prozess durchzuführen ist. Weil die Experimentdaten aus der Produktionslinie gesammelt werden und die meisten Ziffern der Firmendaten keine numerische Bedeutung haben, sind die meisten Daten Zeichen oder Repräsentationen von Kennzahlen und Arbeitsablaufinformationen der Produktionslinien. Ein einfacher Vergleich von unterschiedlichen Attributen der Firmendaten untereinander macht keinen Sinn. Deshalb sollen einige grobe Analyseansätze gesucht werden. Nach der Datenanalyse werden zwei Attribute festgelegt, nach denen die Qualität einzelner Arbeitsabläufe, die in der Hauptdatentabelle durch einzelne „Tag“-Nummern vertreten sind, klassifiziert werden können. Sie sind die Attribute „TotalResult“ und „NmbOfRepairs“. Nach der allgemeinen Logik werden folgende Analyseansätze in dieser Masterarbeit eingesetzt: Wenn der Attributwert des Attributs „TotalResult“ eines Datensatzes „fail“ ist, ist dieser Arbeitsablauf bzw. diese „Tag“-Nummer problematisch, weil dieser Arbeitsablauf zum Schluss durchgefallen ist. Wenn der Attributwert des Attributs „NmbOfRepairs“ eines Datensatzes gleich dem diskretisierten Attributwert „high repair“ ist, ist dieser Datensatz bzw. diese „Tag“-Nummer auch problematisch, weil eine hohe Summe bei der Reparierhäufigkeit gleichzeitig auch hohen Reparieraufwand, Personalaufwand und Zeitaufwand verursacht.

Weil es insgesamt drei Datenbestände gibt und der dritte Datenbestand nicht geeignet für den späteren DM-Prozess ist, sollen die Ergebnisse der Datenvorverarbeitung von den zwei übrigen Datenbeständen extrahiert werden. Das Modell der Datenvorverarbeitung in RapidMiner ist bei beiden Datenbeständen gleich, aber die festgelegten Parameter sind unterschiedlich nach den Daten des jeweiligen Datenbestandes. Die Modellierungsprozesse aus Abschnitt 3.2 werden nun auf Grundlage der Daten vom Datenbestand „AESBig“ erläutert und angezeigt.

In der Abbildung 3.23 werden zwei Balkendiagrammen von der Statistik der Attribute „TotalResult“ und „NmbOfRepairs“ von beiden Datenbeständen mithilfe eines Balkendiagramms gezeigt.

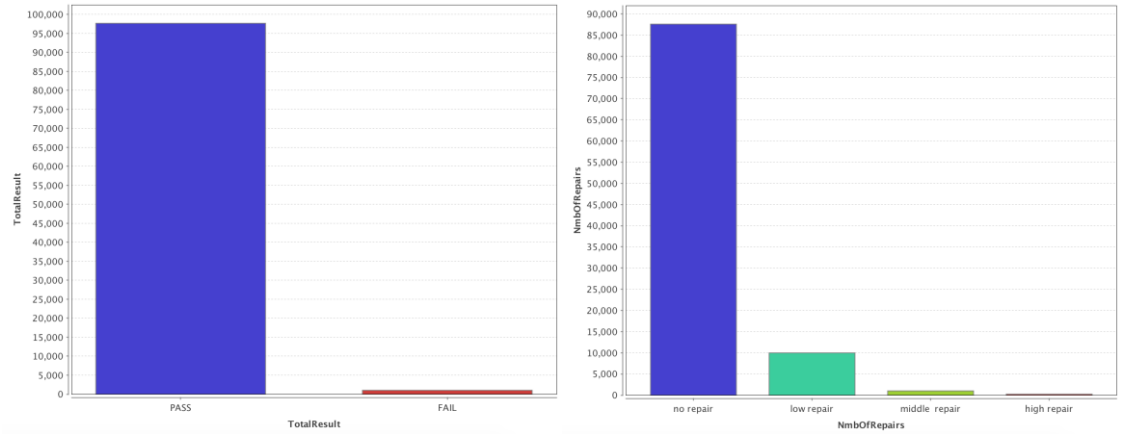


Abbildung 3.23: Statistik der Attribute „TotalResult“ und „NmbOfRepairs“ (nach RapidMiner)

Aus der Grafik der Abbildung 3.23 wird klar, dass die Attributwerte „fail“ und „high repair“ den geringsten Anteil des jeweiligen Attributs belegen und die Prozentzahlen der beiden gering sind.

Innerhalb der vier Haupttabellen hat die Datentabelle „OperationResultProtocol“ den höchsten Detaillierungsgrad. Das heißt, dass die Zahl der Datenzeilen der integrierten HDT identisch zur Anzahl der Datenzeilen der Datentabelle „OperationResultProtocol“ sein soll. Mithilfe von SQL ist diese Summe bekannt, und zwar ungefähr 139 Millionen. Im Vergleich mit dieser Zahl ist die Summe der Datenzeilen (100.000) in der Stichprobe zu gering und beträgt weniger als 0,1 % von 139 Millionen. Wenn die Datenanalyse von zwei problematischen Attributwerten „fail“ und „high repair“ mithilfe einer Stichprobe von 100.000 Datenzeilen durchgeführt wird, ist das Endergebnis ungenau und kann den gesamten Datenbestand kaum repräsentieren. Durch die Analyse der oben gezeigten Screenshots wird herausgefunden, dass die Attributwerte „fail“ und „high repair“ nur einen geringen Prozentsatz in der HDT von beiden Datenbeständen belegen. Entsprechend der Erläuterung am Anfang des Abschnitts 3.4 werden diese beiden Attributwerte als die Analyseansätze zur Beurteilung der Qualität der Attribute betrachtet.

Deshalb wird die Datenanalyse sich nur auf die problematischen Attributwerte „fail“ und „high repair“ konzentrieren. Zusätzlich werden neue numerische Attributwerte beim Attribut „NmbOfRepairs“, deren Anzahl höher als 7 ist, durch die Analyse des vollständigen Datenbestands in SQL herausgefunden. Identisch zu anderen Attributwerten des Attributs „NmbOfRepairs“ werden solche Attributwerte als ein neuer Attributwert „repair>7“ diskretisiert. Die Häufigkeit dieses Attributwertes ist relativ gering im Vergleich anderen Attributwerten des Attributs „NmbOfRepairs“.

Deshalb wird dieser neue Attributwert neben den Attributwerten „fail“ und „high repair“ als der dritte Analyseansatze der Datenanalyse betrachtet.

Um sich in der Datenanalyse vollständig auf die drei ausgewählten Attributwerte konzentrieren zu können, werden zuerst drei Datentabellen mit jeweils 10.000 Datensätzen, deren Attributwerten von den Attributen „TotalResult“ und „NmbOfRepairs“ identisch zu „fail“, „high repair“ und „repair>7“ sind, mithilfe von SQL exportiert. Das genaue Exportverfahren ist identisch zu dem von den Top 100.000 Datenzeilen und dieses Verfahren wurde schon am Anfang des Abschnitts 3.2.1 erläutert. Aber die genauen Befehle sind unterschiedlich. In Anhang 4 werden die benötigen Befehle zum Export von entsprechenden Datentabellen angezeigt.

Nach dem Export der benötigten Datentabellen können die Daten mithilfe des Modells der Datenvorverarbeitung in RapidMiner vorverarbeitet werden und das genaue Modellierungsverfahren wurde bereits oben in den Abschnitten 3.2 und 3.3 erläutert. In späteren Abschnitten wird die Datentabelle, in der sich nur Attributwerte des Attributs „TotalResult“ mit „fail“ befinden, zu FAIL-Datentabelle (FDT) umbenannt. Nach den gleichen Gedanken werden die anderen zwei Datentabellen als „high repair-Datentabelle“ (HRDT) und „NmbOfRepairs more than 7-Datentabelle“ (More7RDT) umbenannt.

Nun beginnt die Interpretation der Ergebnisse der Datenvorverarbeitung.

Die Ergebnisse werden nach den folgenden Gedanken interpretiert: Zuerst werden die relativen Häufigkeiten (RH) der einzelnen Attributwerte jedes Attributes von der FAIL-Datentabelle mit den relativen Häufigkeiten der einzelnen Attributwerte jedes Attributes von der HDT (Hauptdatentabelle mit 10.000-Datenzeilen) mithilfe von Vergleichstabellen verglichen. Die Höhe der RH wird in Form von Prozentzahlen in der Vergleichstabelle angezeigt. Wenn die RH eines Attributwertes zwischen unterschiedlichen Datentabellen große Unterschiede bezüglich der Summe oder Proportion zeigt, wird dieser Attributwert als problematisch betrachtet.

Weil die Summe der Datenzeilen der kompletten integrierten Hauptdatentabelle groß ist, werden in dieser Masterarbeit Stichproben daraus genommen. Um den gesamten Datenbestand besser zu repräsentieren, wird die RH jedes einzelnen Attributwertes jedes Attributes von der HDT in der Vergleichstabelle mit dem Durchschnittswert der RH von den „Top 100.000-HDT“ und „Last 100.000-HDT“ ermittelt. Identisch zur Erläuterung in Abschnitt 3.1.4 ist das Balkendiagramm ein nützliches Werkzeug zur Visualisierung von Ergebnissen. Mit dessen Hilfe können die absoluten Häufigkeiten von allen Attributwerten eines Attributes innerhalb einer Grafik angezeigt werden. Um die Ergebnisse der Datenvorverarbeitung übersichtlich visuell anzeigen zu können, werden die Balkendiagramme von der RH-Verteilung jedes allen Attributen von den „Top 100.000 Datenzeilen“-HDT und „Last 100.000 Datenzeilen“-HDT, FDT, HRDT und More7RDT erstellt. Diese Balkendiagramme werden in pdf-Format exportiert und auf CD gespeichert zur Gewährleistung einer guten Bildqualität. Im Folgenden werden nun die Ergebnisse der einzelnen Attribute anhand der FDT, HRDT und More7RDT in zwei Abschnitten, nämlich „FAIL-Analyse“ und „High repair-Analyse“, hintereinander angezeigt und interpretiert.

FAIL-Analyse

In diesem Abschnitt wird die RH jedes Attributwertes aus der FDT mit der RH jedes Attributwertes von der Hauptdatentabelle mit 10.000 Datenzeilen nach jedem Attribut verglichen. Wenn die RH eines Attributwertes der FDT größer als in der HDT ist, haben die Arbeitsabläufe bzw. „Tag“-Nummern, die diesen Attributwert haben, eine relativ hohe fail-Wahrscheinlichkeit. Wenn die RH eines Attributwertes der FDT geringer als die der HDT ist, haben die Arbeitsabläufe bzw. „Tag“-Nummern mit diesem Attributwert relativ geringe fail-Wahrscheinlichkeit. Am Anfang jedes Attributes wird eine Vergleichstabelle zum Vergleich der RH jedes Attributwertes zwischen den FDT und der „HDT mit 10.000 Datenzeilen“ erstellt.

ManufacturingTime

In der Tabelle 3.4 werden die RH von allen Attributwerten des Attributs „ManufacturingTime“ zwischen den FDT und HDT verglichen.

Tabelle 3.4: FAIL-Analyse des Attributs „ManufacturingTime“

	FDT	HDT		FDT	HDT
<5 min	0	12,4%	30-60 min	13,1%	3,1%
5-10 min	0	39,7%	1-2 h	57,5%	3,4%
10-15 min	0	28,3%	> 2 h	29,4%	2,2%
15-30 min	0	10,9%			

Durch den Vergleich in der Tabelle 3.4 wird herausgefunden: Die Arbeitsabläufe werden am Ende nur durchfallen, wenn ihre Produktionszeit mehr als 30 min beträgt. Das heißt, dass hohe Produktionszeiten auch hohe fail-Wahrscheinlichkeit bedeuten.

NmbOfRepairs

In der Tabelle 3.5 werden die RH von allen Attributwerten des Attributs „NmbOfRepairs“ zwischen den FDT und HDT verglichen.

Tabelle 3.5: FAIL-Analyse des Attributs „NmbOfRepairs“

	FDT	HDT		FDT	HDT
no repair	0	89,7%	high repair	2%	0,2%
low repair	90,3%	9%	repair>7	0,2%	0,1%
middle repair	7,5%	1%			

Durch den Vergleich in der Tabelle 3.5 werden folgende Ergebnisse herausgefunden: Die Arbeitsabläufe, die zu den Attributwerten „low repair“, „middle repair“, „high repair“ gehören, haben eine hohe fail-Wahrscheinlichkeit. Die Arbeitsabläufe, die zum Attributwert „repair>7“ gehören, haben relativ geringere fail-Wahrscheinlichkeit als die obengenannten drei Attributwerte. Es ist auch besonders hervorzuheben, dass die Arbeitsabläufe ohne Reparieren gar nicht durchfallen. Das heißt, dass die Arbeitsabläufe ohne Reparieren stabil sind. Nur wenn ein Arbeitsablauf überhaupt ein Reparierprotokoll hat, soll er besonders beachtet werden.

LineId

In der Tabelle 3.6 werden die RH von allen Attributwerten des Attributs „LineId“ zwischen den FDT und HDT verglichen.

Tabelle 3.6: FAIL-Analyse des Attributs „LineId“

	FDT	HDT
line 2	33,4 %	35,6 %
line 3	32,7 %	30,3 %
line 4	33,9 %	34,1 %

Durch den Vergleich der RH von den FDT und HDT in der Tabelle 3.6 wird herausgefunden, dass die Arbeitsabläufe, die zur „line 3“ gehören, höhere fail-Wahrscheinlichkeit haben als die Arbeitsabläufe, die zu den anderen zwei Linien gehören. Auf sie soll mehr aufgepasst werden.

ParameterDescriptionId

In der Tabelle 3.7 werden die RH von allen Attributwerten des Attributs „ParameterDescriptionId“ zwischen den FDT und HDT verglichen.

Tabelle 3.7: FAIL-Analyse des Attributs „ParameterDescriptionId“

ParameterDescriptionId	FDT	HDT
PDES00000164	8%	8,2%
PDES00000179	8%	8,2%
PDES00000137	3,3%	2,3%
Id 28, 40 – 46, 48, 50, 51, 65, 67, 68, 70, 86, 91 – 93, 95, 97, 98, 106, 110, 114, 121, 122, 130, 143, 144, 158, 160, 161, 165 – 168, 171, 173, 175, 176	1,2% Id 67, 86, 91 – 93, 97, 98, 106, 110, 114, 121, 122, 130, 143, 144, 160, 161 , 166, 168, 175, 176 (1,1%)	1,2%
Id 27, 63, 64, 87, 88, 111, 113, 124, 125, 127, 128, 131, 142, 146, 150, 152, 153, 159, 162, 163, 169, 170, 174, 177	1,1% Id 63, 159, 174 (1,2%) Id 150, 152 (1,0%)	1,1%
Id 71, 112, 148, 149, 151, 178	1,0% Id 71, 112 (1,1%)	1,0%

Durch den Vergleich in der Tabelle 3.7 wird es herausgefunden: Die RH jedes Attributwerts der beiden Datentabellen ist ungefähr gleich. Nur die Id-Nummer „PDES00000137“ ist speziell, weil die RH der fail-Datentabelle dieses Attributwertes um 1 % höher als die der HDT liegt. Die kleinen Unterschiede der RH zwischen der FDT und HDT werden in der Zellen der FDT gelistet. Insgesamt wird es zusammengefasst: Die fail-Wahrscheinlichkeiten von den Arbeitsabläufen, die zu unterschiedlichen ParaDesId gehören, sind ungefähr gleich. Es gibt keine problematischen Attributwerte, deren Arbeitsabläufe speziell hohe fail-Wahrscheinlichkeit haben.

ProductId

In der Tabelle 3.8 werden die RH von allen Attributwerten des Attributs „ProductId“ zwischen den FDT und HDT verglichen.

Tabelle 3.8: FAIL-Analyse des Attributs „ProductId“

	FDT	HDT		FDT	HDT
PROD00000006	38,8%	39,5%	PROD00000011	1,9%	2,6%
PROD00000009	8,3%	8,8%	PROD00000012	28,4%	27,6%
PROD00000010	21,5%	20,8%	PROD00000014	1,2%	0,7%

Durch den Vergleich der RH zwischen den FDT und HDT in der Tabelle 3.8 werden folgende Ergebnisse herausgefunden: Es gibt keine großen Unterschiede in den RH zwischen beiden Datentabellen. Im Vergleich mit anderen Produkten haben die Arbeitsabläufe, die „Product10“,

„Product12“ und „Product14“ zugeordnet werden, relativ höhere fail-Wahrscheinlichkeit als die Arbeitsabläufe von anderen Produkten. Besonders soll „Product14“ beachtet werden, weil dessen Erhöhungsproportion der RH im Vergleich mit der RH der HDT fast 50 % beträgt und diese Anzahl relativ höher als bei anderen Produkten ist.

ResultSequence

In der Tabelle 3.9 werden die RH von allen Attributwerten des Attributs „ResultSequence“ zwischen den FDT und HDT verglichen.

Tabelle 3.9: FAIL-Analyse des Attributs „ResultSequence“

	FDT	HDT		FDT	HDT		FDT	HDT
RS1	10,1%	9,3%	RS8	6,6%	6,7%	RS15	2,4%	2,3%
RS2	7,9%	8%	RS9	6,7%	6,8%	RS16	2,2%	2,2%
RS3	7,9%	8%	RS10	6,7%	6,9%	RS17	1,2%	1,2%
RS4	8%	8%	RS11	5,5%	5,5%	RS18	1,3%	1,2%
RS5	6,8%	6,9%	RS12	4,5%	4,5%	RS19	0,1%	<0,1%
RS6	6,8%	6,9%	RS13	4,6%	4,7%	RS20	0,1%	<0,1%
RS7	6,8%	6,9%	RS14	3,4%	3,4%	RS21-39	<0,1%	<0,1%

Die Daten in der Tabelle 3.9 zeigen, dass die Unterschiede in den RH jedes Attributwertes zwischen beiden Datentabellen nicht groß sind. Das heißt: Die fail-Wahrscheinlichkeiten von den Arbeitsabläufen, die zu unterschiedlichen „ResultSequence” gehören, sind ungefähr gleich und je höher die RH einer „ResultSequence“-Id ist, desto höher ist auch ihre fail-Wahrscheinlichkeit. Im Vergleich mit anderen Attributwerten ist der RH-Unterschied vom Attributwert „RS1“ relativ höher. Das heißt: Die Arbeitsabläufe, die dem „RS1“ zugeordnet werden, haben relativ höhere fail-Wahrscheinlichkeit als die Arbeitsabläufe, die zu den anderen „RSId“ gehören.

RoutingSequence

In der Tabelle 3.10 werden die RH von allen Attributwerten des Attributs „RoutingSequence“ zwischen den FDT und HDT verglichen.

Tabelle 3.10: FAIL-Analyse des Attributs „RoutingSequence“

	FDT	HDT		FDT	HDT
RouSe 10	4,4%	4,5%	RouSe 60	11,1%	11,3%
RouSe 20	21,8%	21,2%	RouSe 70	12%	12,4%
RouSe 30	16%	15,9%	RouSe 80	1,1%	1,1%
RouSe 40	18,1%	18,6%	RouSe 115	1,1%	0,1%
RouSe 50	14,4%	14,9%			

Es ist einfach herauszufinden, dass die Arbeitsabläufe, die zum Attributwert „RoutingSequence115“ gehören, große fail-Wahrscheinlichkeit haben. Zusätzlich ist die RH vom Attributwert „RoutingSequence20“ der FDT um 0,6 Prozentpunkte höher als die der HDT und die Arbeitsabläufe, die diesem Attributwert zugeordnet werden, haben relativ höhere fail-Wahrscheinlichkeit als die anderen RoutingSequenceId außer „RouSe115“.

WorkSequence

In der Tabelle 3.11 werden die RH von allen Attributwerten des Attributs „WorkSequence“ zwischen den FDT und HDT verglichen.

Tabelle 3.11: FAIL-Analyse des Attributs „WorkSequence“

	FDT	HDT		FDT	HDT		FDT	HDT
WS 1	4,5%	4,5%	WS 8	14,2%	11,7%	WS 15	0,4%	0,1%
WS 2	1,1%	19,9%	WS 9	11,2%	1,9%	WS 16	0,4%	0,1%
WS 3	0	15,1%	WS 10	11,1%	0,8%	WS 17	0,2%	<0,1%
WS 4	19,7%	17,3%	WS 11	2%	0,9%	WS 18	0,2%	0,1%
WS 5	14,4%	0,8%	WS 12	1,4%	0,4%	WS 19-26	≤ 0,1%	≤ 0,1%
WS 6	16%	14,4%	WS 13	1,2%	0,4%			
WS 7	1,1%	11,3%	WS 14	0,6%	0,3%			

Der Vergleich der RH zwischen den FDT und HDT in der Tabelle 3.11 zeigt, dass die Ergebnisse von allen Attributwerten in vier Sorten unterteilt werden können. Die erste Sorte heißt „Extrem hohe fail-Wahrscheinlichkeit“. „WorkSequences“ von dieser Sorte haben zwar geringe RH in der HDT, aber sie haben hohe RH in der FDT. „WS 5“, „WS 9“ und „WS 10“ gehören zu dieser Sorte. Die zweite Sorte heißt „Hohe fail-Wahrscheinlichkeit“. Bei dieser Sorte sind die RH aus der FDT deutlich höher als die aus der HDT. „WS 4“, „WS 6“, „WS 8“, „WS 11“, „WS 12“, „WS 13“, „WS 14“, „WS 15“, „WS 16“, „WS 17“, „WS 18“ und „WS 22“ sind Beispiele dafür. Die dritte Sorte heißt „Geringe fail-Wahrscheinlichkeit“. „WorkSequences“ von dieser Sorte haben zwar hohe RH in der HDT, aber sie haben geringe RH in der FDT. „WS 2“, „WS 3“ und „WS 7“ gehören dazu. Diese Sorte ist das Gegenteil von der ersten Sorte. Die vierte Sorte heißt „Normale fail-Wahrscheinlichkeit“. Die RH von beiden Datentabellen bei dieser Sorte sind ungefähr gleich. Die übrigen „WorkSequences“ gehören zu dieser Sorte. Die Arbeitsabläufe, die zu den „WSId“ der ersten und zweiten Sorte gehören, haben höhere fail-Wahrscheinlichkeiten als die Arbeitsabläufe, die zu anderen „WSId“ gehören. Der Zustand von Arbeitsabläufen, die zu den „WSId“ der dritten Sorte gehören, ist genau umgekehrt zu den Arbeitsabläufen, die zu ersten zwei Sorten gehören. Die Arbeitsabläufe, die zu den „WSId“ aus der vierten Sorte gehören, haben mittlere fail-Wahrscheinlichkeit.

WorkPlaceId

In der Tabelle 3.12 werden die RH von allen Attributwerten des Attributs „WorkPlaceId“ zwischen den FDT und HDT verglichen.

Tabelle 3.12: FAIL-Analyse des Attributs „WorkPlaceId“

	FDT	HDT		FDT	HDT
WP 80	4,4%	4,5%	WP 93	14,4%	14,9%
WP 81	21,8%	21,2%	WP 102	11,1%	11.3%
WP 82	16%	15,9%	WP 103	12%	12,3%
WP 91	18,1%	18,6%	WP 113	2,2%	1,2%

Gemäß der Tabelle 3.12 sind die Unterschiede in den RH zwischen beiden Datentabellen nicht groß, aber die Arbeitsabläufe, die dem „WP103“ zugeordnet werden, haben relativ höhere fail-Wahrscheinlichkeit als andere Arbeitsabläufe, die zu anderen „WPId“ gehören.

Remarks

In der Tabelle 3.13 werden die RH von allen Attributwerten des Attributs „Remarks“ zwischen den FDT und HDT verglichen.

Tabelle 3.13: FAIL-Analyse des Attributs „Remarks“

	FDT	HDT		FDT	HDT
Safety check: PASS	20,6%	20,4%	Safety check: FAIL	1,2%	0,7%
HV+EC test: PASS	17,9%	18,5%	Calibration: FAIL	0,1%	0,2%
Manual test: PASS	15,7%	15,8%	Manual test: FAIL	0,3%	0,2%
Calibration: PASS	14,3%	14,6%	HV+EC test: FAIL	0,2%	0,2%
Flashing 2: PASS	11,9%	12,2%	Flashing 1: FAIL	0,2%	0,1%
Flashing 1: PASS	10,9%	11,3%	Laser (9999999999IO-Teile): PASS	1,1%	0,1%
AESStart: PASS	0	4,5%	Flashing 2: FAIL	<0,1%	0,1%
Marking: PASS	1,1%	1,1%	AESStart: FAIL	4,4%	<0,1%

Die Ergebnisse dieses Attributes werden aus theoretischem und praktischem Gesichtspunkt nacheinander interpretiert. Theoretisch haben die Arbeitsabläufe, die Attributwerte „AESStart: FAIL“, „Safety check: FAIL“, „Laser (9999999999IO-Teile): PASS“, „Manual test: FAIL“ und „Flashing 1: FAIL“ haben, relativ höhere fail-Wahrscheinlichkeit als andere Arbeitsabläufe, die zu anderen „Remarks“ gehören, weil die RH der FDT von diesen fünf Attributwerten höher als die der HDT und die Erhöhungsausmaße der RH nach dem Gesichtspunkt der Proportion höher als die der anderen Attributwerte sind. Besonders sind die Attributwerte „AESStart: FAIL“ und „Laser (9999999999IO-Teile): PASS“ zu nennen. Sonst sind die RH-Unterschiede von anderen Attributwerten nicht groß. Das heißt: Die Arbeitsabläufe, die zu solchen Attributwerten gehören, haben mittlere fail-Wahrscheinlichkeit. Nach dem praktischen Gesichtspunkt wird jeder Attributwert mit den Optionen PASS oder FAIL markiert. Die Attributwerte des Attributs „Remarks“, die mit PASS markiert werden, zeigen die höchste RH in beiden Datentabellen. Eigentlich sollen die Arbeitsabläufe, die mit PASS beim Attribut „Remarks“ markiert werden, zum Ende auch bestehen. Aber die Tatsache ist nicht identisch zu der Vermutung. Aber grundsätzlich sind die fail-Wahrscheinlichkeiten von den Arbeitsabläufen, die den Attributwert „pass“ zeigen, niedriger als die, denen der Attributwert „fail“ zugeordnet wird.

ProcessId

In der Tabelle 3.14 werden die RH von allen Attributwerten des Attributs „ProcessId“ zwischen den FDT und HDT verglichen.

Tabelle 3.14: FAIL-Analyse des Attributs „ProcessId“

	FDT	HDT
Process 1	95,6%	95,5%
Process 99	4,4%	4,5%

Beim Vergleich zeigt sich, dass die fail-Wahrscheinlichkeiten von den Arbeitsabläufen, die zu beiden Prozessen gehören, in beiden Tabellen ungefähr gleich sind.

Aggregiertes Attribut

In Abschnitt 3.2.2 wurden die inneren Zusammenhänge zwischen den Attributen „WorkSequence“, „RoutingSequence“, „WorkplaceId“, „ProcessId“ und „Remarks“ schon genau erläutert. Nach der Analyse der Datentabelle „OperationProtocol“ gibt es insgesamt 11 unterschiedliche normale aggregierte Attributwerte von den obengenannten fünf Attributen und sie werden in der Tabelle 3.15 zusammen mit ihren RH aus der HDT aufgelistet.

Tabelle 3.15: Normale aggregierten Attributwerte

Normale aggregierte Attributwerte	Relative Häufigkeit in der HDT
WS 1 + RouSe 10 + WP 80 + Process 99 + AESStart: PASS	4,5%
WS 2 + RouSe 20 + WP 81 + Process 01 + SafetyCheck: PASS	18,2%
WS 3 + RouSe 30 + WP 82 + Process 01 + ManualTest: PASS	14,9%
WS 4 + RouSe 40 + WP 91 + Process 01 + HV+ES test: PASS	16,9%
WS 5 + RouSe 45 + WP 92 + Process 01 + Switch ON NTM: PASS	0
WS 6 + RouSe 50 + WP 93 + Process 01 + Calibration: PASS	13,5%
WS 7 + RouSe 60 + WP102 +Process 01 + Flashing 1: PASS	10,3%
WS 8 + RouSe 70 + WP103 +Process 01 + Flashing 2: PASS	11,3%
WS 9 + RouSe 80 + WP113 +Process 01+ Making: PASS	1%
WS10+ RouSe 85 + WP120 +Process 01+ DMC-Check: PASS	0
WS11+ RouSe 90 + WP121 +Process 01+ Robot: PASS	0
Summe	90,6%

Zwar erscheinen nicht alle aggregierten Attributwerte in der HDT, aber die oben gezeigten normalen aggregierten Attributwerte können offensichtlich die meisten Fälle repräsentieren, indem ihre Summe 90,8% beträgt.

Nachfolgend wird die Tabelle 3.16 von den aggregierten Attributwerten erstellt, die hohe RH in der FDT haben. Weil es zahlreiche unterschiedliche aggregierten Attributwerte gibt, werden nur solche aggregierten Attributwerte in der Tabelle angezeigt, derer RH mehr als 1 % beträgt.

Tabelle 3.16: FAIL-Analyse des Attributs „Aggregiertes Attribut“

Aggregierte Attributwerte	Relative Häufigkeit
WS 4 & RouSe20 & WP81 & Process1 & Safety check: PASS	18,5%
WS 6 & RouSe40 & WP91 & Process1 & HV+EC test: PASS	15,9%
WS 5 & RouSe30 & WP82 & Process1 & Manual test: PASS	14,2%

WS 8 & RouSe50 & WP93 & Process1 & Calibration: PASS	12,7%
WS 10 & RouSe70 & WP103 & Process1 & Flashing 2: PASS	10,6%
WS 9 & RouSe60 & WP102 & Process1 & Flashing 1: PASS	9,7%
WS 1 & RouSe10 & WP80 & Process99 & AESStart: FAIL	4,4%
WS 4 & RouSe20 & WP81 & Process1 & Safety check: FAIL	1,1%
WS 2 & RouSe115 & WP113 & Process1 & Laser (9999999999IO-Teile): PASS	1,1%
WS 7 & RouSe20 & WP81 & Process1 & Safety check: PASS	1,1%
WS 11 & RouSe80 & WP113 & Process1 & Marking: PASS	1%

Ergebnisse der Datenanalyse sind folgende Punkte:

1. Kein aggregierter Attributwert der FDT, dessen RH mehr als 1 % beträgt, ist einer der normalen aggregierten Attributwerte, die in der Tabelle 3.15 stehen.
2. Die meisten aggregierten Attributwerte, deren RH mehr als 1% in der FDT beträgt, sind mit PASS markiert.
3. Das Teilattributwert des Attributs „Remarks“ hängt vom Teilattributwert des Attributes „WorkplaceId“ ab.

„High repair“-Analyse

Im letzten Abschnitt wurden die Datenvorverarbeitungsergebnisse bezüglich des „TotalResult“-Attributwerts „fail“ mithilfe der Vergleichstabelle ausführlich gezeigt und interpretiert. In diesem Abschnitt wird nun das gleiche Verfahren auf den Vergleich der relativen Häufigkeiten jedes Attributwertes von allen Attributen zwischen drei Datentabellen angewendet, nämlich „Hauptdatentabelle mit 10.000 Datenzeilen“ (HDT), „High repair“-Datentabelle (HRDT) und „NmbOfRepairs more than 7“-Datentabelle (More7RDT). Das Ziel dieser Datenanalyse ist, die Attributwerte herauszufinden, die hohe „high repair“-Wahrscheinlichkeit haben. Die konkrete Vergleichstabelle jedes Attributes wird im Anhang 13 angezeigt.

ManufacturingTime

Der Vergleichstabelle der RH „High repair“-Analyse des Attributs „ManufacturingTime“ im Anhang 13 zeigt: Die Reparierhäufigkeit eines Arbeitsablaufs beträgt nur mehr als 4, wenn seine Produktionszeit mehr als eine Stunde dauert. Aber Arbeitsabläufe mit einer Produktionszeit von mehr als einer Stunde machen insgesamt nur 5,6 % aller Arbeitsabläufe aus. Wenn die Produktionszeit eines Arbeitsabläufe mehr als 2 Stunden dauert, ist ihre „high repair“-Wahrscheinlichkeit relativ hoch.

TotalResult

Der Vergleichstabelle der RH in der Tabelle „High repair“-Analyse des Attributs „TotalResult“ im Anhang 13 zeigt: Die „fail“-RH von HRDT und More7RDT sind höher als die der HDT. Deshalb wird die Kenntnisse erlangt, dass es die fail-Wahrscheinlichkeit eines Arbeitsablaufs deutlich erhöht, wenn seine Reparierhäufigkeit mehr als vier beträgt.

LineId

Beim Vergleich der RH in der Tabelle „High repair“-Analyse des Attributs „LineId“ im Anhang 13 wird herausgefunden, dass die Arbeitsabläufe, deren Reparierhäufigkeit mehr als 4 beträgt, sich vor allem auf die Linien 3 und 4 konzentrieren. Die RH von beiden Linien in HRDT und

More7RDT sind jeweils ungefähr gleich. Aber im Vergleich mit ihren RH in der HDT zeigt sich, dass die Arbeitsabläufe in der Linie 3 höhere „high repair“-Wahrscheinlichkeit als die von Linie 4 haben.

ParameterDescriptionId

Im Anhang 13 werden die RH von allen Attributwerten des Attributs „ParameterDescriptionId“ zwischen den HRDT, More7RDT und HDT durch die Tabellen „High repair“-Analyse des Attributs „ProductId“(1) und „High repair“-Analyse des Attributs „ProductId“(2) verglichen. Weil die Ergebnisse von drei Datentabellen große Unterschiede zueinander haben, werden die HDT mit anderen zwei Datentabellen nacheinander verglichen.

Der Vergleich der RH in beiden Tabellen zeigt: PDES00000164 und PDES00000179 haben die höchste RH in der HDT. Gleichzeitig zeigen die beiden Attributwerte auch die höchste RH in HRDT und More7RDT. Aber ihre RH sind in HRDT und More7RDT um fast 1% reduziert im Vergleich zu der RH in der HDT. Unterschiedlich zu den Ergebnissen der FAIL-Analyse ist die Verteilung der RH von den übrigen aggregierten Attributwerten nicht durchschnittlich, die nicht identisch wie die normalen aggregierten Attributwerte sind. In beiden Tabellen im Anhang 13 werden die Attributwerte angezeigt, deren RH mehr als 1,1 % beträgt. Neben den Attributwerten „164“ und „179“ haben die beiden Datentabellen viele gemeinsame Attributwerte. Das heißt, dass die Arbeitsabläufe, die diesen ParaDesId zugeordnet werden, hohe „high repair“-Wahrscheinlichkeit haben.

ProductId

Durch Vergleich der RH in der Tabelle „High repair“-Analyse des Attributs „ProductId“ im Anhang 13 zwischen drei Datentabellen werden folgende Ergebnisse erhalten: Die Arbeitsabläufe, die „Product 9“ und „Product 10“ zugeordnet werden, haben eine höhere „high repair“-Wahrscheinlichkeit als die anderen Arbeitsabläufe, die zu anderen ProductId gehören. Deshalb müssen die Arbeitsabläufe, die zu „Product10“ und „Product 9“ gehören, besonders beachtet werden. Zusätzlich hat die Reparierhäufigkeit von den Arbeitsabläufen von Produkt14 hohe Wahrscheinlichkeit, mehr als 7 zu betragen, indem die RH der More7RDT um 40 % größer als die der HDT ist.

ResultSequence

Der Vergleich der RH in der Tabelle „High repair“-Analyse des Attributs „ResultSequence“ im Anhang 13 zeigt: Von „RS1“ bis „RS 10“ sind die RH der HDT höher als die von den HRDT und More7RDT. Das heißt: Die Arbeitsabläufe, die RS1 bis RS10 zugeordnet sind, haben geringe „high repair“-Wahrscheinlichkeit. Aber für die Arbeitsabläufe, die zu „RS19“ bis „RS36“ gehören, ist es umgekehrt. Die RH von solchen „RSId“ sind zwar nicht hoch, aber ihre Erhöhungsproportion im Vergleich mit den RH der HDT ist deutlich. Das heißt, dass die Arbeitsabläufe, die diesen „RSId“ zugeordnet werden, hohe „high repair“- Wahrscheinlichkeit haben. Für die übrigen „RSId“ sind die RH aus den Tabellen HRDT und More7RDT zwar auch höher als die der HDT, aber die Erhöhungsproportion ist nicht so hoch wie „RS19“ bis „RS36“. Das heißt: Die Arbeitsabläufe, die zu „RS11“ bis „RS18“ gehören, haben eine mittlere „high repair“-Wahrscheinlichkeit.

RoutingSequence

Der Vergleich in der Tabelle „High repair“-Analyse des Attributs „RoutingSequence“ im Anhang 13 ergibt: Die Arbeitsabläufe, die den „RouSe10“, „RouSe 20“ und „RouSe115“ zugeordnet werden, haben höhere „high repair“-Wahrscheinlichkeit als andere „RoutingSequences“, weil ihre RH in HRDT und More7RDT deutlich höher als die in der HDT sind. Die RH-Unterschiede sind relativ klein bei „RouSe30“. Das heißt, dass die Arbeitsabläufe, die „RouSe30“ zugeordnet, mittlere „high repair“-Wahrscheinlichkeit haben. Die übrigen „RouSeId“ sind gute „RouSe“, weil deren RH in HRDT und More7RDT deutlich geringer als die in der HDT sind. Das heißt, dass die Arbeitsabläufe, die zu solchen „RouSeId“ gehören, geringe „high repair“-Wahrscheinlichkeit haben.

WorkSequence

Durch den Vergleich in der Tabelle „High repair“-Analyse des Attributs „WorkSequence“ im Anhang 13 wird herausgefunden, dass dieses Attribut identisch zu den Ergebnissen der FAIL-Analyse besonders beachtet werden sollte. Aber die genauen Ergebnisse sind unterschiedlich im Vergleich mit der FAIL-Analyse. Bei der „High repair“-Analyse sind die RH von HRDT, More7RDT jeder „WorkSequence“ höher als 1 %, wobei die RH in der HDT von manchen „WorkSequences“ weniger als 1 % betragen. Die oben gezeigten Ergebnisse sind nach dem Gesichtspunkt „Unterteilungskriterien“ ähnlich wie die Ergebnisse der FAIL-Analyse. Manche „WorkSequences“ sind problematisch. Ihre RH in HRDT und More7RDT sind deutlich höher als die in der HDT, nämlich „WS 1“, „WS 5“, „WS 10-26“. Das heißt: Die Arbeitsabläufe, die solchen WorkSequenceId zugeordnet sind, haben relativ hohe „high repair“-Wahrscheinlichkeit. Manche „WorkSequences“ sind gut. Ihre RH in HRDT, More7RDT sind deutlich geringer als die in HDT, nämlich „WS 2-4“, „WS 6-8“. Das heißt: Die Arbeitsabläufe, die solchen WorkSequenceId zugeordnet sind, haben relativ geringe „high repair“-Wahrscheinlichkeit. Die RH von den problematischen „WorkSequences“ werden mit roter Farbe markiert.

WorkPlaceId

Durch den Vergeich der RH in der Tabelle „High repair“-Analyse des Attributs „WorkplaceId“ im Anhang 13 wird gezeigt: Identisch zu letzten Attribut „WorkSequence“ können die Ergebnisse dieses Attributes auch in zwei Gruppen unterteilt werden. Die „WPId“ in der ersten Gruppe sind problematisch. Ihre RH in HRDT und More7RDT sind deutlich höher als die von der HDT, nämlich „WP 81“ und „WP 91“. Das heißt: Die Arbeitsabläufe, die solchen „WorkplaceId“ zugeordnet werden, haben höhere „high repair“-Wahrscheinlichkeit als andere „WorkplaceId“, die zu anderen „Remarks“ gehören. Die „WPId“ in der ersten Gruppe sind gut. Ihre RH in HRDT und More7RDT sind deutlich geringer als die von der HDT, nämlich „WP 80“, „WP 102“ und „WS 103“. Das heißt: Die Arbeitsabläufe, die diesen „WorkplaceId“ zugeordnet werden, haben geringere „high repair“-Wahrscheinlichkeit als die von anderen „WorkplaceId“, die zu anderen „Remarks“ gehören. Zusätzlich gibt es bei diesem Attribut noch die dritte „Normale Gruppe“. Ihre RH in HRDT und More7RDT sind ungefähr identisch zu der RH in der HDT, nämlich für „WP 82“, „WP93“ und „WP 113“. Das heißt: Die entsprechenden Arbeitsabläufe haben mittlere „high repair“-Wahrscheinlichkeit.

Remarks

Durch den Vergeich der RH in der Tabelle „High repair“-Analyse des Attributs „Remarks“ im Anhang 13 wird gezeigt:letzten Attribut können die verschiedenen „Remarks“ in drei Gruppen unterteilt werden. Die erste „Problematische Gruppe“ enthält folgende „Remarks“: „Safety check: FAIL“, „Calibration: FAIL“, „HV+EC test: FAIL“, „Manual test: FAIL“, „Flashing 1: FAIL“, „Flashing 2: FAIL“ und „Laser (9999999999IO-Teile): PASS“. Das heißt: Die Arbeitsabläufe, denen solchen „Remarks“ zugeordnet werden, haben höhere „high repair“-Wahrscheinlichkeit als andere Arbeitsläufe, die zu anderen „Remarks“ gehören, weil ihre RH in HRDT und More7RDT deutlich höher als die in der HDT sind. Die zweite „Gute Gruppe“ enthält folgende „Remarks“: „Manual test: PASS“, „Calibration: PASS“, „Flashing 1: PASS“, „Flashing 2: PASS“, „AESStart: PASS“ und „Marking: PASS“. Das heißt: Die Arbeitsabläufe, denen solche „Remarks“ zugeordnet sind, haben geringere „high repair“-Wahrscheinlichkeit als andere Arbeitsabläufe mit anderen „Remarks“. Die dritte „Normale Gruppe“ enthält folgende „Remarks“: „Safety check: PASS“, „HV+EC test: PASS“ und „AESStart: FAIL“. Das heißt: Die Arbeitsabläufe mit solchen „Remarks“ haben mittlere „High repair“-Wahrscheinlichkeit. Es wird herausgefunden, dass die meisten „Remarks“ von der „Problematischen Gruppe“ mit FAIL markiert werden, während die meisten „Remarks“ der „Guten Gruppe“ und der „Normalen Gruppe“ ein PASS zeigen. Der Attributwert „Laser (9999999999IO-Teile): PASS“ ist aber eine Ausnahme und auf seine Arbeitsabläufe soll besonders aufgepasst werden, weil er zur „Problematischen Gruppe“ gehört, aber mit PASS markiert wird.

ProcessId

Aus dem Vergleich in der Tabelle „High repair“-Analyse des Attributs „ProcessId“ im Anhang 13 zeigt sich: Die Arbeitsabläufe, die dem „Process99“ zugeordnet werden, haben relativ geringere „high repair“-Wahrscheinlichkeit als die Arbeitsabläufe von „Process1“.

Aggregiertes Attribut

Identisch zur Erklärung im Abschnitt „Datenaggregation“ und in der FAIL-Analyse bekannt, gibt es 11 normale aggregierte Attributwerte, die die höchste Häufigkeit im neuen aggregierten Attribut der HDT belegen. Weil die Summe der aggregierten Attributwerte zu hoch ist, werden nur die aggregierten Attributwerte angezeigt, deren RH nicht unter 1 % beträgt in der Tabelle „High repair“-Analyse von „Aggregierten Attributen“ im Anhang 13 gelistet. Aggregierte Attributwerte, die identisch zu normalen aggregierten Attributwerten sind, werden mit roter Farbe hervorgehoben.

Durch die Datenanylse der Tabelle „High repair“-Analyse von „Aggregierten Attributen“ im Anhang 13 werden folgenden Ergebnisse herausgefunden:

1. Beide Datentabellen HRDT und More7RDT haben aggregierte Attributwerte, die identisch zu den normalen aggregierten Attributwerten sind. Zwar beträgt die Summe nur 4, aber diese vier aggregierte Attributwerte belegen die höchste Häufigkeit in jeder oben gezeigten Datentabelle. Deshalb haben die Arbeitsabläufe, die zu diesen vier aggregierten Attributwerten gehören, relativ hohe „high repair“-Wahrscheinlichkeit.
2. Die meisten aggregierten Attributwerte, die in der Vergleichstabelle stehen, sind keine normalen aggregierten Attributwerte. Deshalb haben die Arbeitsabläufe, die nicht zu den nor-

malen aggregierten Attributwerte gehören, höhere „high repair“-Wahrscheinlichkeit als die Arbeitsabläufe, die zu den normalen aggregierten Attributwerten gehören.

3. Die aggregierten Attributwerte mit hohen RH werden fast alle mit PASS markiert. Das heißt, dass ein Wert PASS keine Sicherheit bedeutet. Umgekehrt sollen solche entsprechenden Arbeitsabläufe im späteren Prozess besonders beachtet werden.
4. HRDT und More7RDT haben viele gemeinsame aggregierte Attributwerte, die RH sind jedoch unterschiedlich.
5. Die ersten fünf aggregierten Attributwerte, welche die höchsten RH der jeweiligen Datentabelle belegen, werden mit blauer Farbe markiert, und die entsprechenden Arbeitsabläufe sollen besonders beachtet werden. Davon ist nur „WS 2 & RouSe20 & WP81 & Process1 & Safety check: PASS“ ein gemeinsamer aggregierter Attributwert von den HRDT und More7RDT und seine RH belegen den ersten Platz in beiden Datentabellen. Deshalb haben die Arbeitsabläufe, die diesem Attributwert zugeordnet werden, relativ hohe „high repair“- Wahrscheinlichkeit.

3.5 Fazit

In diesem Kapitel wurden drei Datenvorverarbeitungsmethoden auf die Firmendaten angewendet, nämlich Datenhomogenisierung, Datenaggregation und Feature Selection. Die Struktur dieses Kapitels ist grundsätzlich entsprechend dem Vorgehensmodell zur MESC aufgebaut, das vom Lehrstuhl IT in Produktion und Logistik entwickelt wurde. Zuerst wurde die Aufgabenstellung dieser Masterarbeit vorgestellt, es wurden Modellierungsdaten ausgewählt, die Zusammenhänge zwischen unterschiedlichen Datentabellen analysiert und die angewendete Software kurz vorgestellt. Danach begann der Datenvorverarbeitungsprozess. Vier Haupttabellen wurden nach den Analyseergebnissen des ER-Modells ausgewählt und mithilfe von gemeinsamen Attributen durch die Software SQL zu einer Hauptdatentabelle integriert und die integrierte Hauptdatentabelle wurde als das Zielformat der Datenbestände betrachtet und ihr Inhalt als Inputdaten für den Datenvorverarbeitungsprozess eingesetzt. In Abschnitt 3.2.1 wurden Attribut-Konstruktion und Attribut-Extraktion durchgeführt, mit denen die originale HDT entsprechend dem Bedarf des späteren DM-Prozesses erweitert wird. Um die Anzahl der Attributwerte zu reduzieren, werden die originalen Attributwerte nach unterschiedlichen Aggregationsstufen durch Diskretisierung in neuen Attributwerten aggregiert. Mithilfe der Chi-Square-Statistik wurden die Attribute nach ihrer Relevanz mit dem Attribut „Tag“ ausgewählt, auf das sich nach der Aufgabenstellung im Modellierungsprozess konzentriert werden soll.

Nach der Durchführung von drei Datenvorverarbeitungsmethoden wird die originale HDT komprimiert und die redundanten Daten werden gefiltert. Anschließend werden zwei Attribute als die Analyseansätze für die Datenanalyse ausgewählt, nämlich „TotalResult“ und „NmbOfRepairs“. Die drei Attributwerte „FAIL“, „High repair“ und „repair>7“ unter diesen werden für die Produktion als kritisch angesehen und drei Datentabellen, deren Attributwerte von den Attributen „TotalResult“ und „NmbOfRepairs“ identisch zu oben genannten drei problematischen Attributwerten sind, werden speziell zu ihrer besseren Analyse aus der Datenbank exportiert. Das Ziel ist, die Attributwerte von anderen Attributen herauszufinden, deren zugehörige Arbeitsabläufe hohe Wahrscheinlichkeit haben, am Ende durchzufallen oder bei denen die Summe an Reparaturen mehr als 4 beträgt. Die Datenanalyse wird in Form einer Vergleichstabelle realisiert. Nachfolgend werden die Ergebnisse der Datenvorverarbeitung zusammengefasst.

Durch die Vergleichstabellen zwischen den FDT und HDT wurde herausgefunden: Die Unterschiede in der RH von Attributwerten zwischen FDT und HDT konzentrieren sich auf folgende Attribute: „ManufacturingTime(Second)“, „NmbOfRepairs“ und „WorkSequence“. Das heißt: Die Wahrscheinlichkeit, dass ein Arbeitsablauf am Ende durchfällt, ist höher, wenn seine Produktionszeit mehr als 30 min beträgt, er mindestens eine Reparatur braucht oder er zu den WS mit den Id-Nummern 5 und 9 bis 26 gehört. Durch die Attribut-Konstruktion werden fünf Attribute, die innere Zusammenhänge miteinander haben, zu einem Attribut aggregiert. Die aggregierten Attribute, deren RH in der FDT hoch ist, werden in der Tabelle 3.4.14 aufgelistet. Solche Attribute dienen auch dafür, die Gründe für ein Durchfallen herauszufinden.

Durch Vergleich zwischen den Tabellen HRDT, More7RDT und HDT wird herausgefunden: Die RH von fast allen Attributwerten in HRDT und More7RDT ist unterschiedlich im Vergleich mit der RH der Attributwerte in der HDT. Die genauen Analyseergebnisse werden in dieser Masterarbeit mithilfe von drei Formen interpretiert, nämlich in schriftlicher Form, Tabellenform und Grafikform. Hier wird die genaue Interpretation nicht wiederholt. Die gemeinsamen Merkmale der Unterschiede in der RH von allen Attributen in HRDT und More7RDT sind folgende: Manche Attributwerte, die eine geringe RH in der HDT zeigen, zeigen eine hohe RH in HRDT und More7DT. Manche Attributwerte, die eine hohe RH in der HDT zeigen, haben hingegen eine geringe RH in HRDT und More7DT. Das heißt: Solche Attributwerte, deren RH in HRDT und More7DT im Vergleich mit der RH in der HDT erhöht sind, haben eine höhere Wahrscheinlichkeit als die anderen Attributwerte, dass ihre Reparaturhäufigkeit mehr als vier beträgt.

Es wird zusammengefasst, dass die RH der Attributwerte von unterschiedlichen Attributen in der FDT zwar keine großen Unterschiede in ihrer RH gegenüber der HDT haben, die Verteilung der RH der Attributwerte in HRDT und More7DT jedoch große Unterschiede im Vergleich mit der HDT hat. Das heißt: Die Gründe für ein Durchfallen von Produkten nach den Arbeitsabläufen sind relativ schwer zu finden und das Durchfallen ist normal in der Produktion, obwohl andere Attributwerte statt dem aktuellen Attributwert angewendet werden, weil die Durchfalls-Wahrscheinlichkeiten von unterschiedlichen Attributwerten in vielen Attributen ungefähr gleich sind. Der Zustand ist aber umgekehrt beim Attribut „NmbOfRepairs“ und auch die Durchfalls-Wahrscheinlichkeit von unterschiedlichen Attributwerten ist in vielen Attributen verschieden. Das heißt: Die Attributwerte, die eine hohe Wahrscheinlichkeit haben, dass ihre Reparaturhäufigkeit mehr als vier beträgt, sind relativ einfach dadurch zu finden. Damit können mögliche Gründe für eine hohe Reparaturhäufigkeit von manchen Arbeitsabläufen durch die Analyse der Vergleichstabellen herausgefunden werden.

4. Anwendung des Clusterverfahrens auf die Firmendaten

In diesem Abschnitt wird entsprechend der Fragestellung ein geeignetes Data Mining- Verfahren für die Firmendaten gesucht, um zu Fragestellung mithilfe dieses Verfahrens die gewünschten Kenntnisse zu extrahieren. Zuerst werden in Abschnitt 4.1 die Vorbereitungsschritte des Clusterverfahrens erläutert. Anschließend in Abschnitt 4.2 wird der Modellprozess des Clusterverfahrens anhand von zwei angewendeten Algorithmen erläutert, die im Abschnitt 2.4.5 schon erläutert wurden. Nach der Durchführung der Modellprozesse werden die Ergebnisse nach den Validierungsverfahren evaluiert, die in Abschnitt 2.4.6 bereits dargestellt wurden. Danach werden die Analyseergebnisse im Abschnitt 4.3 weiterverarbeitet zur Extraktion der handlungsrelevanten Ergebnisse. Die Darstellungsform der DM-Ergebnisse wird zur einer geeigneten Form überführ, um die Ergebnisse besser zu interpretieren. Im letzten Abschnitt 4.4 wird das Fazit aus den Modellprozessen und den DM-Ergebnissen gezogen.

4.1 Vorbereitung der Clusteranalyse

Nach dem Vorgehensmodell von MESC von [ITP16] sollen folgende Aufgaben, die durch die Tabelle 4.1.1 gezeigt werden, in diesem Abschnitt bearbeitet werden.

Tabelle 4.1: Aufgabendefinition für die Vorbereitung des DM-Verfahrens (nach [ITP16])

4. Vorbereitung des Data Mining-Verfahrens	4.1 Verfahrenswahl	Auswahl des einzusetzenden Verfahrens in Abhängigkeit von der Aufgabenstellung
	4.2 Werkzeugauswahl	Auswahl eines geeigneten Data Mining- Werkzeugs
	4.3 Fachliche Kodierung	Fachliche Auswahl und Kodierung geeigneter Attribute
	4.4 Technische Kodierung	Technische Auswahl und Kodierung geeigneter Attribute

In diesem Abschnitt wird zuerst die Anwendung der Clusteranalyse begründet. Anschließend werden zwei genaue Algorithmen der Clusteranalyse nach dem Zustand der Firmendaten ausgewählt. Zum Schluss werden die notwendigen „fachlichen Kodierungs"- und „technischen Kodierungs"-Prozesse durchgeführt, um das Clusteranalyse durchführen zu können.

4.1.1 Verfahrens- und Werkzeugauswahl

Durch die Datenanalyse am Anfang des Kapitels 3 wurde deutlich, dass die ursprünglichen Firmendaten ungeordnet und komplex sind. Die Firma hat keinen genaue Analyseansatz festgelegt. Deshalb wird entschieden, das Clusteranalyse als DM-Verfahren einzusetzen, weil es ein *unsupervised*-DM-Verfahren ist. Das heißt, dass keine bestimmten Kriterien vor dem DM-Prozess manuell festgelegt werden müssen. Als Ergebnis werden die versteckten Cluster, also die eventuellen Gruppen, von den Firmendaten erkannt.

Die Auswahl eines geeigneten Clusteranalyse-Algorithmus ist eine wichtige und schwierige Aufgabe. Nach der Vorstellung in Abschnitt 2.4.4 gibt es hauptsächlich drei Clusteranalyse-Methoden, nämlich partitionsmethode, hierarchische Methode und density-basierte Methode.

Jeder Clusteranalyse-Algorithmus wird diesen drei Methoden zugeordnet, z. B. k-Means- und Erwartungsmaximierungs-Algorithmus gehören zur Partitionsmethode, Bottom-up- und Top-down-Algorithmus gehören zur hierarchischen Methode. Das Problem ist, dass die meisten Clusteralgorithmen nur geeignet für numerische Daten sind. Der Datentyp der meisten Firmendaten ist jedoch nominal. Die traditionellen Clusteranalyse-Algorithmen sind aufgrund der Berechnung von Distanzen und Dichten nicht geeignet für nominale Daten [Agg15, S.206]. Durch die zahlreichen Experimente im Rahmen der Clusteranalyse mit den Firmendaten wird entschieden, den EM-Algorithmus als Clusteralgorithmus für die nominalen Firmendaten anzuwenden. [Agg15] bietet eine gute Lösung zur Anwendung der numerischen Clusteralgorithmen auf die nominalen Daten. Sie ist die Transformation der nominalen Daten zu binären nominalen Daten, weil die binären nominalen Daten ein spezieller Fall von numerischen Daten sind. Nach der Transformation des Datentyps können die Clusteralgorithmen für numerische Daten auf die binären nominalen Daten angewendet werden, z. B. k-Mean-Algorithmus. Der Datentyptransformationsprozess wird als „fachliche Kodierung“ genannt und wird im nächsten Abschnitt genau erläutert.

4.1.2 Fachliche Kodierung und technische Kodierung der Firmendaten

Der Prozess der Datentyptransformation wird auch als ein Prozess der fachlichen Kodierung betrachtet und alle Datenwerte des Datasets in diesem Prozess mit den Werten „0“ oder „1“ kodiert werden. Der Prozess „Technische Kodierung“ wird im Modell durchgeführt, um die fehlenden und verrauschten Daten mit der Ziffer „9999999999“ zu kodieren und später zu filtern. Dieser Prozess ist genau identisch zu dem im Modellprozess der Datenvorverarbeitung.

Funktionsweise

Durch die Analyse der Datenvorverarbeitungsergebnisse wird gesehen, dass jedes Attribut viele unterschiedliche Attributwerte haben. Nach der Datentyptransformation wird ein neues Attribut für jeden Attributwert jedes originalen Attributes in der neuen Datentabelle von binären Daten erstellt. Somit wird die originale Datentabelle zu einer hochdimensionalen Datentabelle transformiert. Die Bestimmung des Wertes jedes Datenfeldes in der binären Datentabelle hängt von der Hauptfrage ab, ob der Attributwert eines Attributs bei einem bestimmten Attributwert des Attributs „Tag“ erscheint. Wenn der Attributwert erscheint, wird in das entsprechende Datenfeld in der neuen Datentabelle der binären Daten der Wert „1“ eingegeben, sonst wird der Wert „0“ eingegeben. In der Abbildung 4.1 wird ein Beispiel der Ergebnisse vom Prozess „Fachliche Kodierung“ mithilfe eines Screenshots gezeigt.

Tag	NmbOfRep...	N...	N...	...	ProcessId ...	Pr...	LineId = li...	Li...
152123601...	1	0	0	0	0	1	0	1
152123601...	1	0	0	0	1	0	0	1
152123601...	1	0	0	0	1	0	0	1
152123601...	1	0	0	0	1	0	0	1
152123601...	1	0	0	0	1	0	0	1
152123601...	1	0	0	0	1	0	0	1

Abbildung 4.1: Beispiel des Ergebnisses der „Fachliche Kodierung“ (nach RapidMiner)

Durch die Analyse der Datenvorverarbeitungsergebnisse wird herausgefunden, dass die Zahl an Attributwerten von den meisten Attributen hoch ist. Deshalb ist es nicht möglich, ein neues Attribut für jeden Attributwert der originalen Attribute zu erstellen. Deshalb ist die Aggregation der Attributwerte der originalen Attribute notwendig. Nach dem Modellprozess im Abschnitt 3.2.1 ist die Methode „Diskretisierung“ eine gute Lösung für diesen Fall.

Ein wichtiger Schwerpunkt der Diskretisierung ist das Gleichgewicht der Skalierung jedes neuen Attributwertes. Im Abschnitt 3.4 „Visualisierung und Interpretation der Ergebnisse“ werden die relativen Häufigkeiten (RH) jedes Attributwertes nach unterschiedlichen Attributen aus der HDT in entsprechenden Vergleichstabellen zusammengefasst. Es ist häufig der Fall, dass zwei Attributwerte, die sich nebeneinander in der originalen Reihenfolge befinden, große Unterschiede bei der RH haben. Zum Beispiel: Das Attribut „ProductId“ hat 9 Attributwerte, nämlich „Product 1“ bis „Product 9“. Die RH jedes Attributwertes wird mithilfe der Tabelle 4.2 in Form einer Prozentzahl angezeigt.

Tabelle 4.2: Beispielprozess der Aggregation der Attributwerte eines Attributes (nach RapidMiner)

ProductId	RH (%)	ProductId	RH (%)	ProductId	RH (%)
Product 1	2%	Product 4	10%	Product 7	3%
Product 2	20%	Product 5	3%	Product 8	5%
Product 3	30%	Product 6	12%	Product 9	15%

Aus der Tabelle 4.2 wird klar, dass die prozentuale Verteilung nicht homogen ist. So ist z. B. der Unterschied der RH zwischen „Product 1“ und „Product 2“ groß, obwohl sich beide „ProductId“ nach der Reihenfolge der Id-Nummern nebeneinander befinden. Wenn der Datentyp direkt transformiert würde, würden 8 neue Datentabellen erstellt. Das ist jedoch zu viel. Wenn die Attributwerte einfach nach der Reihenfolge der Id-Nummern aggregiert würden, ist die Aggregationsskalierung nicht homogen und die Ergebnisse des späteren DM-Prozess können verfälscht werden, wenn „Product 1 bis 3“, „Product 4 bis 6“ und „Product 7 bis 9“ jeweils eine Gruppe bilden. Wenn „Product 1“ ein großes Problem hat und „Product 2“ und „Product 3“ keine Probleme haben, wäre dann das Problem von „Product 1“ aber schwer zu entdecken, weil „Product 1“ eine relative geringe RH mit der ersten Gruppe belegt. Um die Ergebnisse des späteren DM-Prozesses nicht zu verfälschen, sollen die unterschiedlichen Attributwerte nach der Höhe ihrer RH aggregiert werden. In dieser Masterarbeit werden die alten Attributwerte „Product 1“ bis „Product 9“ nach dem Kriterium „Höhe der RH“ zu drei neuen Attributwerten aggregiert, näm-

lich „high frequency“, „middle frequency“ und „low frequency“. Später werden diese drei Kriterien mit den Namen „hFre“, „mFre“ und „lFre“ abgekürzt. Nach diesem Aggregationsverfahren sind „Product 2, 3“ dem neuen Attributwert „hFre“ zuzuordnen, „Product 4, 6, 9“ dem neuen Attributwert „mFre“ und „Product 1, 5, 7, 8“ dem neuen Attributwert „lFre“.

Die genauen RH der einzelnen Attributwerte jedes Attributs in der HDT wurden bereits in Abschnitt 3.4 mithilfe der Vergleichstabellen gezeigt. Nach den RH der einzelnen Attributwerte werden drei neue Attributwerte für jedes originale Attribut erstellt. Das Aggregationsergebnis der Attributwerte wird mithilfe einer Datentabelle in Anhang 14 angezeigt. Im späteren DM-Prozess wird der k-Means-Algorithmus auf die fachlich kodierten Daten angewendet. Damit die Summe der Attribute nach dem „fachlichen Kodierungs“-Prozess nicht zu hoch wird und die RH der Attributwerte in unterschiedlichen aggregierten Gruppen homogen sind, werden die Attribute der originalen Datentabellen nach dem oben gezeigten Verfahren aggregiert

Um den gesamten Datenbestand besser zu repräsentieren, wird noch eine weitere Stichprobe genommen, wie oben in Abschnitt 3.1.2 beschrieben. Es soll besonders aufgepasst werden, dass die Summe der Attributwerte des Attributs „NmbOfRepairs“ nicht immer gleich ist. Durch die Datenanalyse der weiteren Datentabellen wurde herausgefunden, dass dieses Attribut sich in 5 Attributwerte unterteilt, nämlich „no repair“, „low repair“, „middle repair“, „high repair“ und „repair>7“. Der letzte Attributwert existiert nicht in der Top 100.000-HDT und soll zusätzlich bei der Datenanalyse von den anderen Stichproben berücksichtigt werden.

Modellierung in RapidMiner

Um den Datentyp der Firmendaten von „nominal“ bis „binär“ zu transformieren bzw. den „fachlichen Kodierungs“-Prozess durchführen zu können, wird ein Modell in RapMin aufgebaut. Der Modellprozess der Datentyptransformation wird mithilfe von drei Unterprozessen aufgebaut, nämlich „Diskretisierung der nominalen Daten“, „Diskretisierung der numerischen Daten“ und „Datentyptransformation“. Im Folgenden werden diese drei Unterprozesse vorgestellt.

Diskretisierung der nominalen Daten

Zunächst wird der Modellprozess „Diskretisierung der nominalen Daten“ in der Abbildung 4.2 durch ein Screenshot gezeigt.

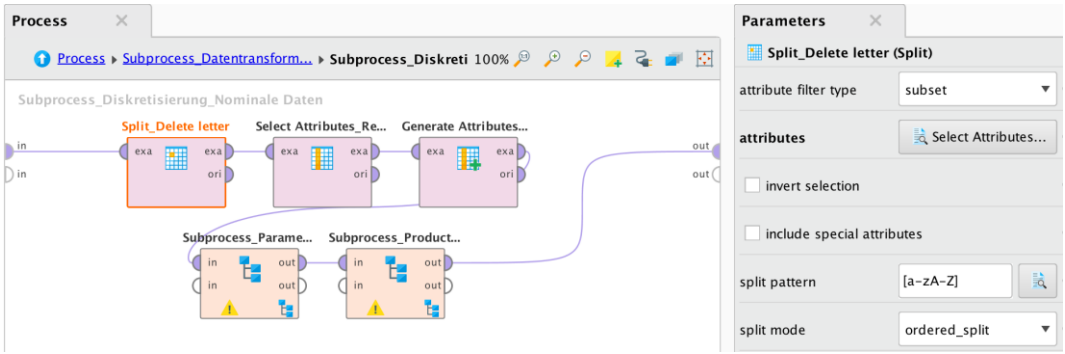


Abbildung 4.2: Modellprozess „Diskretisierung der nominalen Daten“ (nach RapidMiner)

Der erste Unterprozess ist die Diskretisierung der nominalen Daten, weil der Operator „Discretize“ in RapidMiner nur für die numerischen Daten geeignet ist. Deshalb werden einige spezielle Methoden angewendet, um die Methode „Diskretisierung“ für die nominalen Daten realisieren zu können. Es soll besonders aufgepasst werden, dass aus technischen Gründen von RapidMiner nur die nominalen Attribute, die Ziffern enthalten, mit den folgenden Methoden diskreti-

siert werden können. Nach der Datenanalyse wird entschieden, die Attribute „ProductId“ und „ParameterDescriptionId“ mithilfe dieser Methode zu diskretisieren. Der Modellprozess beginnt mit dem Operator „Split“. In diesem Operator wird ein „split pattern“ festgelegt, um die Buchstaben in den Datenfeldern zu bereinigen. Der Parameter „split mode“ wird als „ordered_split“ festgelegt. Der Operator „Split“ produziert redundante Attribute und diese werden mithilfe des Operators „Select Attributes“ bereinigt. Nach Durchführung der ersten zwei Operatoren sind die Daten schon numerische Daten, aber der Datentyp ist noch unverändert. Um den Datentyp zu transformieren, wird der Operator „Generate Attributes“ angewendet, der bei der Modellierung des Datenvorverarbeitungsmodells auch eingesetzt wird. In diesem Operator wird eine Funktion *„parse()“* auf die Datentyptransformation angewendet. Danach beginnt der Diskretisierungsprozess. In der Abbildung 4.3 wird der Modellprozess der „Diskretisierung“ des Attributs „ProductId“ als ein Beispiel gezeigt.

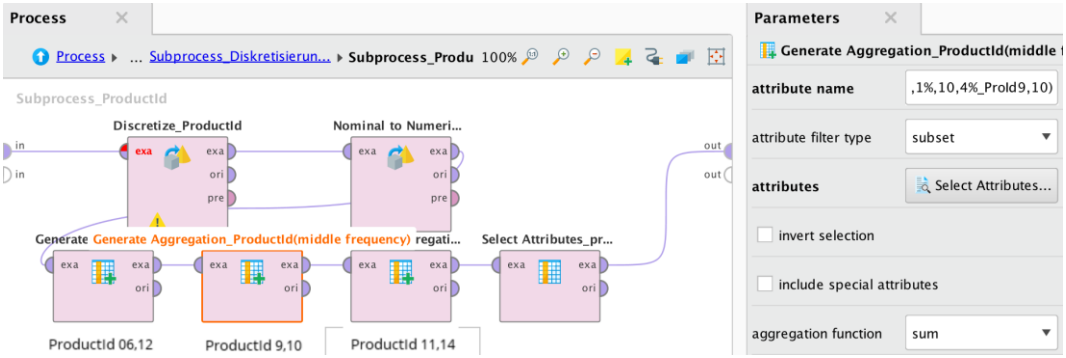


Abbildung 4.3: Beispiel-Diskretisierungsprozess des Attributs „ProductId“ (nach RapidMiner)

Weil die Attributwerte des Attributs „ProductId“ nach der Datentyptransformation nur Ziffern umfassen und nicht identisch dum originalen Datenformat „Product6“ sind, werden sie mithilfe des Operators „Discretize“ nach dem originalen Datenformat neu benannt. Dieser Prozess ist identisch zu dem im Abschnitt „Datenhomogenisierung“. Um den Datentyp wieder zu „numerical“ transformieren, wird der Operator „Nominal to Numerical“ angewendet. Dieser Operator wurde auch vorher im Abschnitt „Datenhomogenisierung“ zur Datentyptransformation eingesetzt. Aber für den Parameter *coding type* wird in diesem Fall die Option „dummy code“ gewählt und nicht die Option *unique integers* wie beim Modell der Datenhomogenisierung. Der Unterschied zwischen den beiden Parametern ist: Mithilfe des Parameters *dummy code* wird ein neues Attribut für jeden Attributwert des alten Attributs erstellt. Die genaue Funktionsweise dieses Prozesses wurde schon am Anfang des Abschnitts 4.1.2 erklärt und somit nicht nochmals wiederholt. Bei der Einstellung des Parameters *unique integers* wird kein neues Attribut erstellt. Das heißt, dass die originalen Attributwerte unverändert bleiben. Nur der Datentyp wird transformiert. Nach der Durchführung des Operators „Nominal to Numerical“ werden große Summe von neue Attribute erstellt, die den Rechenaufwand des späteren DM-Prozess stark erhöhen.

Um die Summe an Attributen zu reduzieren, wird der Operator „Generate Aggregation“ eingesetzt. Identisch zu der Erklärung am Anfang dieses Abschnittes werden die Attributwerte jedes originalen Attributs zu drei neuen Attributwerten aggregiert, nämlich „hFre“, „mFre“ und „lFre“. Der Parameter *aggregation function* wird mit der Option *sum* festgelegt, weil die Attributwerte von binären Daten nur „0“ und „1“ sind. Die produzierten redundanten Attribute werden mithilfe des Operators „Select Attributes“ bereinigt.

Diskretisierung der numerischen Daten

In diesem Prozess werden die Attribute in zwei Gruppen unterteilt. Die Attributwerte von den Attributen der ersten Gruppe sollen nicht aggregiert werden, weil die Zahl der Attributwerte von solchen Attributen gering ist, z. B. „lineId“. Im Gegensatz dazu sollen die Attributwerte von der zweiten Gruppe aggregiert werden, weil deren Summe an Attributwerten hoch ist, z. B. „Work-Sequence“. Der Modellprozess von den Attributen der ersten Gruppe wurde bereits in Abschnitt 3.2.1 vorgestellt. Der Modellprozess von Attributen der zweiten Gruppe ist fast identisch zu dem von den nominalen Daten „ProductId“ und „ParaDesId“. Somit werden die beiden Modellprozesse nicht nochmals gezeigt.

Datentyptransformation

In diesem Unterprozess werden die Attribute von den numerischen Daten, deren Datentyp im letzten Unterprozess nicht transformiert wurde, mithilfe des Operators „Nominal to Numerical“ transformiert und kodiert. Der Datentyp des Attributs „Tag“ wird mithilfe des Operators „Numerical to Polynominal“ transformiert. Solche Operatoren wurden schon in Abschnitt 3.2.2 vorgestellt und ebenfalls nicht wiederholt.

4.2 Modellierung

In diesem Abschnitt werden zwei Clusteranalyse-Modelle nach zwei unterschiedlichen Clusteralgorithmen aufgebaut. Die angewendeten Algorithmen wurden bereits in Abschnitt 2.4.5 vorgestellt. Nach dem Vorgehensmodell von MESC von [ITP16] sollen folgende Aufgaben, die durch Tabelle 4.3 gezeigt werden, in diesem Abschnitt erledigt werden.

Tabelle 4.3: Aufgabendefinition zur Vorbereitung des Clusteranalyse-Verfahrens nach [ITP16]

5. Anwendung der DM-Verfahren	5.1 Entwicklung eines Data Mining-Modells	Modellentwicklung und Trennung der Datenbestände in Trainings-, Validierungs- und Testdaten
	5.2 Training des Data Mining-Modells	Training des Data-Mining-Modells mittels Validierung aus 5.1

Nach „fachlichen Kodierung“ können die Clusteralgorithmen für numerische Daten auf die kodierten binären Daten angewendet werden. Zuerst wird in Abschnitt 4.2.1 der k-Means-Algorithmus auf die kodierten binären Daten angewendet. Danach wird in Abschnitt 4.2.2 der EM-Algorithmus auf das Ergebnis der Datenvorverarbeitung angewendet, also auf die nominalen Daten. In Abschnitt 4.2.3 werden die Ergebnisse der zwei Clusteralgorithmen evaluiert und verglichen.

4.2.1 k-Means-Algorithmus

Nach der theoretischen Erklärung in Abschnitt 2.4.5 ist der k-Means-Algorithmus ein Centroid-basierter Clusteralgorithmus. Deshalb soll zuerst die Summe der k-Werte im Cluster festgelegt werden. Anschließend sollen die Parameter *measure types* und *measure* nach dem Datentyp bestimmt werden. Zum Schluss sollen die Parameter *max optimization steps* und *max runs* nach bestimmten Beurteilungskriterien und durch zahlreiche Versuche festgelegt werden. Nach der

Einstellung von Parametern wird das Modell aufgebaut und kann durchlaufen werden. Durch die Abbildung 4.4 wird der Modellprozess dargestellt:

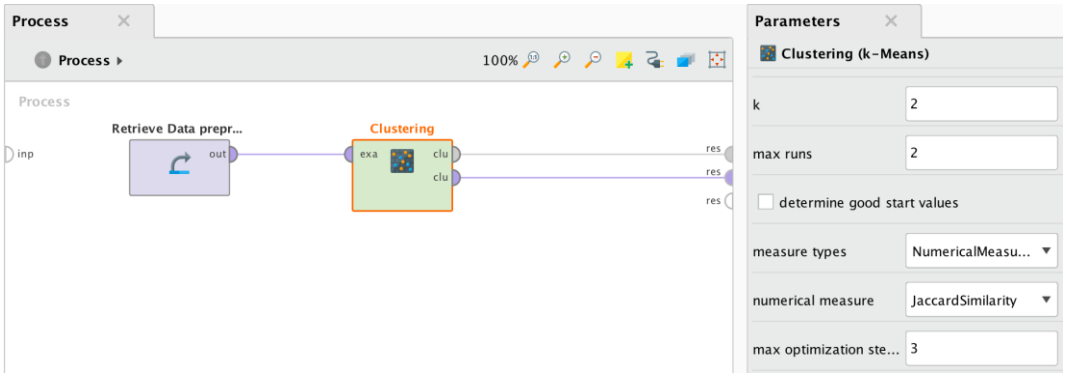


Abbildung 4.4: Modellprozess vom k-Means-Algorithmus (nach RapidMiner)

Dieses Modell ist einfacher als die letzten Modelle. Das Ergebnis von „fachlichen Kodierung“ wird mithilfe des Operators „Retrieve“ als Inputdaten eingesetzt. Danach wird die Clusteranalyse mithilfe des Operators „Clustering (k-Means)“ durchgeführt.

Zuerst soll der Wert des Parameters *k* festgelegt werden. Um die Cluster bezüglich der drei Häufigkeit-Ausmaße, nämlich „high frequency“, „middle frequency“ und „low frequency“, zu generieren, wird für den Parameter *k* der Wert 3 festgelegt. Nach dem Ergebnis der Datenvorverarbeitung wird entschieden: Um die Cluster bezüglich der unterschiedlichen diskretisierten Attributwerte vom Attribut „NmbOfRepairs“, also „no repair“, „low repair“, „middle repair“, „high repair“ und „NmbOfRepairs>7“, zu generieren, wird der Parameter k dafür mit dem Wert 4 oder 5 belegt. Bei der HDT mit Top 100.000 Datenzeilen wird als dieser Parameter k der Wert 4 festgelegt, weil der Attributwert „NmbOfRepairs>7“ bei der HDT nicht existiert. Um die Cluster bezüglich der Attributwerte vom Attribut „TotalResult“, nämlich pass und fail, zu generieren, wird wegen nur zwei Alternativen für k der Wert 2 gesetzt.

Danach sollen der Messungstyp und der Messungsalgorithmus bestimmt werden. Weil die binären Daten auch eine spezielle Form von den numerischen Daten sind, wird der Parameter „NumericalMeasures“ ausgewählt. Nach der theoretischen Erklärung im Abschnitt 2.4.3 wird das „Jaccard Similarity“ als Messungsalgorithmus ausgewählt. Nach der Funktionsweise des „k-Means“-Algorithmus, die im Abschnitt 2.4.5 schon genau erklärt wird, werden die optimale Anzahl von den Parametern „max optimization steps“ und „max runs“ erreicht, wenn das Centroid nach einem Optimierungsschritt unverändert bleibt. Das direkte Beurteilungskriterium davon ist der Distanz zwischen dem einzelnen Datenpunkt und dem Centroid-Punkt. Das heißt, dass die Distanzen zwischen jedem Datenpunkt und jeweiligem Centriod jedes Clusters, die im „Centiod table“ stehen, nach einem Optimierungsschritt unverändert bleiben.

Durch das Durchlaufen des Modellprozesses und den Vergleich des „Centroid Table“ – der Centroid-Tabelle – wird die Anzahl der Parameter *max optimization steps* und *max runs* festgelegt. Die Anzahl von Parametern des k-Means-Algorithmus wird nach unterschiedlichen Modellprozessen in einer Tabelle aufgelistet, die im Anhang 15 angezeigt wird.

4.2.2 Erwartungsmaximierungs-Algorithmus

Der EM-Algorithmus ist ein weicher Algorithmus und jedem Datenobjekt wird mit einer bestimmten Wahrscheinlichkeit ein Cluster zugeordnet. Das Ziel ist, den gesamten Erwartungswert für alle Datenobjekte, die einem bestimmten Cluster zugeordnet sind, zu maximieren. In RapidMiner wird der EM-Algorithmus mithilfe des Operators „Expectation maximization“ realisiert. Durch die Abbildung 4.5 wird der Modellprozess als Screenshot gezeigt.

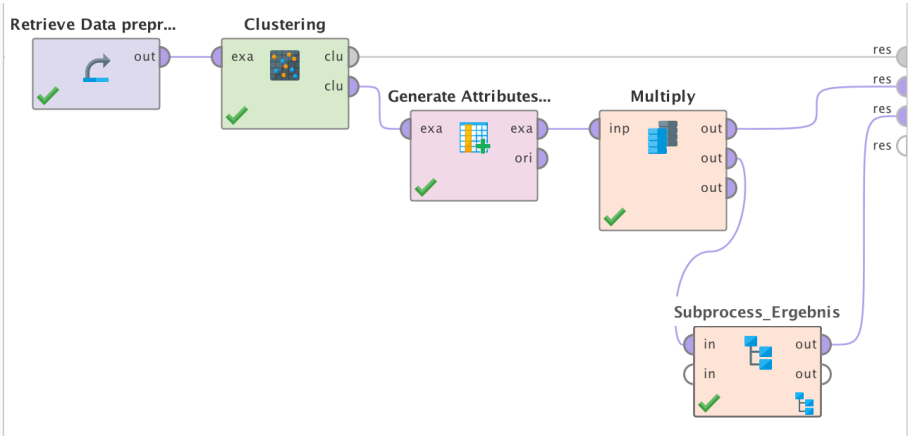


Abbildung 4.5: Modellprozess des EM-Algorithmus (nach RapidMiner)

Die Eingabedaten werden mithilfe des Operators „Retrieve“ importiert. Dann wird die Clusteranalyse mit dem Operator „Expectation maximization“ durchgeführt. Der k-Wert wird mit den gleichen Werten wie im k-Means-Algorithmus festgelegt. Weil das Ziel des EM-Algorithmus die Maximierung des Erwartungswertes ist, werden die Anzahl des Parameters *max run* und die der *max optimization steps* auf Basis zahlreicher Experimente festgelegt, sobald der Erwartungswert maximal ist. Die Anzahl aller Parameter in unterschiedlichen Datentabellen wird durch eine Tabelle gelistet, die im Anhang 17 angezeigt wird. Für den Parameter *quality* wird der vorgegebene Wert „1.0E-10“ festgelegt. Dieser Parameter bedeutet, dass der Algorithmus gestoppt wird, wenn die Anzahl von Nachkommastellen des Ergebnisses 10 überschreitet. Nach der Erläuterung in Abschnitt 2.4.5 wird der Parameter mit der Option *average parameters* belegt. Es soll besonders aufgepasst werden, die Option *correlated attributes* zu kreuzen, weil unterschiedliche Attribute der Firmendaten Zusammenhänge miteinander haben, die schon vorher im Abschnitt 3.1.3 genau erklärt werden. Nach dem EM-Operator wird der endliche Erwartungswert weiter zusammen mithilfe von späteren Operatoren berechnet. Die Berechnungsverfahren wurden schon in Abschnitt 2.4.5 genau erläutert. In RapidMiner wird das Berechnungsverfahren mithilfe des Operators „Generate Attributes“ und „Aggregate“ realisiert. Die Berechnungsformeln ohne Summierung werden mit dem erstgenannten Operator realisiert und die Ergebnisse mithilfe des zweiten Operators summiert, damit der endliche Erwartungswert erhalten wird. In der Abbildung 4.6 wird ein Screenshot der Ergebnisse von beiden Berechnungsprozessen gezeigt.

P(X)	Erwartungswert
0.896	-0.047
0.896	-0.047
0.896	-0.047
0.896	-0.047
0.896	-0.047
0.896	-0.047

Row No.	sum(cluster_0_probability)	sum(cluster_1_probability)	sum(Erwartungswert)
1	94146.356	4605.644	-8620.991

Abbildung 4.6: Ergebnisausgabe der beiden Berechnungsverfahren (nach RapidMiner)

Der Operator „Generate Attributes_Überprüfung“ dient zur Überprüfung der Summen von Datenobjekten, die zu unterschiedlichen Clustern gehören, weil diese Ergebnissumme unterschiedlich zwischen dem Menü „Cluster model“ und „Example Set“ ist. In Anhang 16 werden die Screenshots von allen konkreten Schritten dieses Modellprozesses angezeigt.

4.2.3 Clustervalidierung

Nach der theoretischen Erläuterung in Abschnitt 2.4.6 werden zwei unterschiedliche Kategorien der Clustervalidierungs-Methoden vorgestellt, nämlich interne Clustervalidierungs-Methode und relative Clustervalidierungs-Methode. Im Folgenden wird der Clustervalidierungs-Prozess nach der jeweiligen Clustervalidierungs-Methode und mit den Firmendaten nacheinander erläutert.

Interne Clustervalidierungsmethoden

In dieser Masterarbeit werden die Clusteranalyse-Ergebnisse mithilfe von zwei internen Clustervalidierungsmethoden validiert, nämlich Davies-Bouddin-Index-Methode und Wahrscheinlichkeitsmaß-Methode. Darunter werden die Ergebnisse durch beiden Methoden validiert.

Davies-Bouldin-Index-Methode

Nach der theoretischen Erklärung in Abschnitt 2.4.6 ist diese Methode speziell für den k-Means-Algorithmus geeignet und eine gute Methode zur Bestimmung der Qualität des Clusters. Nun wird der Modellprozess dieser Methode durch die Abbildung 4.7 gezeigt.

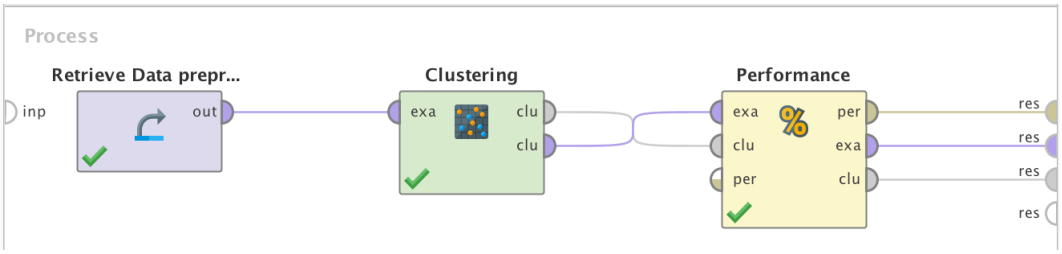


Abbildung 4.7: Modellprozess der Davies-Bouldin-Index-Methode (nach RapidMiner)

Die Validierungsfunktion dieser Methode wird mithilfe des Operators „Performance (Distance)“ realisiert. Nach dem Durchlauf des Modellprozesses wird Davies-Bouldin-Index als Ergebnis bekommen. Die Ergebnisse sind unterschiedlich mit unterschiedlichen Parametereinstellungen im Operator „Clustering“. Die Ergebnisse dieser Methode werden im Anhang 18 gezeigt. Weil die optimalen Werte der Parameter *Max runs* und *Max optimization steps* durch den *k*-Wert festgelegt werden, werden nur die k-Werte zur Unterscheidung der unterschiedli-

chen Parametereinstellungen angezeigt. Nach der theoretischen Erklärung in Abschnitt 2.4.6 ist die Qualität eines Clusters umso höher, je geringer der Davies-Bouldin-Index ist.

Wahrscheinlichkeitsmaß-Methode

Nach der Erklärung in Abschnitt 2.4.6 ist diese Methode geeignet für die Clustervalidierung des EM-Algorithmus. Das Beurteilungskriterium der Qualität eines Clusters nach dieser Methode ist die Höhe des Erwartungswertes, die schon im EM-Modellprozess berechnet wird, wobei der Modellprozess in Abbildung 4.5 schon gezeigt wurde. Die maximalen Erwartungswerte von unterschiedlichen k-Werten werden mithilfe einer Tabelle zusammengefasst im Anhang 19 gezeigt. Identisch zur Davies-Bouldin-Index-Methode werden die optimalen Werte der Parameter *Max runs* und *Max optimization steps* durch den k-Wert festgelegt, deshalb werden nur die k-Werte zur Unterscheidung der unterschiedlichen Parametereinstellungen gezeigt.

Nach der theoretischen Erklärung in Abschnitt 2.4.6 ist die Qualität eines Clusters umso höher, je höher des Erwartungswertes ist. Bei der FDT sind die Erwartungswerte nicht bestimmbar, wenn der k-Wert gleich 3 ist, weil bei irgendeinem Optimierungsschritt und Laufschritt die Anzahl von allen Clustern nicht immer „nicht Null" beträgt. Weil die k-Werte nach dem Datenanalysebedarf festgelegt werden, wenn die Anzahl eines Clusters Null beträgt, ist dieses Clusterergebnis dagegen den Datenanalysebedarf.

Relative Clustervalidierungsmethode

Nach der theoretischen Erklärung im Abschnitt 2.4.6 werden die Clusteranalyse-Ergebnisse von k-Means-Algorithmus und EM-Algorithmus mithilfe der Methode „Fehlerrate" validiert und verglichen. Um die Clusteralgorithmen für die numerische Daten auf die Firmendaten angewendet werden zu können, wird die „fachliche Kodierung" durchgeführt und die Firmendaten werden in drei unterschiedliche relative Häufigkeitsausmaße, nämlich „high frequency", „middle frequency" und „low frequency", aggregiert. Aber das nominale Datenvorverarbeitungsergebnis werden nicht nach den unterschiedlichen Häufigkeitsausmaßen aggregiert. Deshalb werden die Ergebnisse von unterschiedlichen Experimenten mit „k=3" in dieser Masterarbeit nicht miteinander verglichen. Die Ergebnisse von beiden Algorithmen werden entsprechend den k-Werten „2, 4, 5" in der Tabelle 4.4 verglichen. Das Wort „Cluster" wird in diesen Tabellen mit den Buchstaben „CL" verkürzt. Die Größe der Stichprobe jedes Experiments beträgt 100.000.

Tabelle 4.4: Fehlerrate der Ergebnisse von k-Means- und EM-Algorithmen

K=2 („Top 100.000"-Datenzeilen)					
k-Means	Falsche Klassifikation	Fehlerrate	EM	Falsche Klassifikation	Fehlerrate
CL0: FAIL	9757 (10816)	0,09880	CL0: PASS	999 (96780)	0,02948
CL1: PASS	0 (87936)		CL1: FAIL	1912 (1972)	
Stichprobengröße	98752		Stichprobengröße	98752	
K=4 („Top 100.000"-Datenzeilen)					
k-Means	Falsche Klassifikation	Fehlerrate	EM	Falsche Klassifikation	Fehlerrate
CL0: n. b.	n. b. (37951)	n. b.	CL0: mid. repair	2834 (2876)	0,07345

CL1: n. b.	n. b. (22878)		CL1: no repair	3148 (89271)	
CL2: n. b.	n. b. (27295)		CL2: high repair	107 (109)	
CL3: low repair	7688 (10628)		CL3: low repair	1164 (6496)	
Stichproben-größe	98752		Stichproben-größe	98752	
K=2 („Last 100.000“-Datenzeilen)					
k-Means	Falsche Klassifikation	Fehler-rate	EM	Falsche Klassifikation	Fehler-rate
CL0: PASS	847 (93076)	0,06554	CL0: PASS	292 (97403)	0,01049
CL1: FAIL	5626 (5700)		CL1: FAIL	744 (1373)	
Stichproben-größe	98776		Stichproben-größe	98776	
K=5 („Last 100.000“-Datenzeilen)					
k-Means	Falsche Klassifikation	Fehler-rate	EM	Falsche Klassifikation	Fehler-rate
CL0: n. b.	n. b. (18849)	n. b.	CL0: low repair	1456 (10802)	n. b.
CL1: n. b.	n. b. (25294)		CL1: no repair	1979 (81061)	
CL2: n. b.	n. b. (42760)		CL2: n. b.	n. b. (3523)	
CL3: n. b.	n. b. (5575)		CL3: n. b.	n. b. (2263)	
CL4: low repair	n. b. (6298)		CL4: mid. repair	842 (1127)	
Stichproben-größe	98776		Stichproben-größe	98776	
K=4 (FAIL-Datentabelle)					
k-Means	Falsche Klassifikation	Fehler-rate	EM	Falsche Klassifikation	Fehler-rate
CL0: n. b.	n. b. (28613)	n. b.	CL0: mid. repair	7341 (7764)	0,16285
CL1: n. b.	n. b. (25727)		CL1: low repair	8127 (90120)	
CL2: mid. repair	44 (4833)		CL2: repair>7	74 (75)	
CL3: n. b.	n. b. (39841)		CL3: high repair	582 (1055)	
Stichproben-größe	99014		Stichproben-größe	99014	
K=2 (High repair-Datentabelle)					
k-Means	Falsche Klassifikation	Fehler-rate	EM	Falsche Klassifikation	Fehler-rate
CL0: PASS	4834 (59504)	0,39102	CL0: PASS	7102 (93168)	0,07955
CL1: FAIL	31733(34014)		CL1: FAIL	337 (350)	
Stichproben-größe	93518		Stichproben-größe	93518	
K=2 (repair>7-Datenzeilen)					
k-Means	Falsche Klassifikation	Fehler-rate	EM	Falsche Klassifikation	Fehler-rate
CL0: FAIL	13264 (13744)	0,34700	CL0: PASS	1307 (40472)	0,03915

CL1: PASS	908 (27101)		CL1: FAIL	292 (373)	
Stichproben-größe	40845		Stichproben-größe	40845	

Die Zahl in den Klammern jeder Zelle bedeutet die Summe der Datensätze vom Cluster, die in der linken Zelle steht und durch die Clusteranalyse automatisch festgelegt wird. Die Namen von unterschiedlichen Clustern werden nach dem Vergleich zwischen der Summe der Datensätze des Attributwertes in der entsprechenden Datentabelle und der Summe der Datensätze des Clusters festgelegt. So beträgt z. B. beim Experiment der „Top 100.000"-Datentabelle des k-Means-Algorithmus die Summe der Datensätze von Cluster 1 87936. Nach der Statistik über die RH des Attributwertes „PASS" in Abschnitt 3.4.1 beträgt die Summe der Datensätze des Attributwertes „PASS" ungefähr 98000 in der HDT. Deshalb wird für Cluster 1 als Attributwert „PASS" definiert und Cluster 0 wird mit dem anderen Attributwert „FAIL" belegt. In manchen Fällen sind die Cluster nicht bestimmbar, weil die Summe der Datensätze dieses Clusters stark unterschiedlich im Vergleich mit der Summe der Datensätze jedes Attributwertes in der entsprechenden Datentabelle ist. Beispielsweise beim Experiment der „Top 100.000"-Datenzeilen vom k-Means-Algorithmus können nicht alle Cluster definiert werden, weil im Vergleich mit der Summe der Datensätze jedes Attributwertes vom Attribut „NmbOfRepairs" der HDT die Summe der Datensätze jedes Clusters nicht darauf angepasst werden kann. Die nicht bestimmbaren Cluster werden mit dem Zeichen „n. b." gekennzeichnet. Die Zahl in jeder Zelle der Spalte „Falsche Klassifikation" bedeutet die Summe der Datensätze, die falsch zum linken Cluster klassifiziert wird. Die Fehlerrate wird nach der Formel 2.13 durch die beiden Zahlen in jeder Zelle der Spalte „Falsche Klassifikation" berechnet.

Durch den Vergleich in der Tabelle 4.4 wird herausgefunden, dass die Clusteranalyse-Ergebnisse von beiden Clusteralgorithmen in der vorher festgelegten Analyseansatz, z. B. „PASS" und „FAIL" nicht genau übereinstimmen. Nach der relativen Clustervalidierungs- methode sind die Ergebnisse des EM-Algorithmus deutlich besser als die des k-Means-Algorithmus. Aber die Summe der Datensätze der Attributwerte mit geringeren RH in einem Attribut kann meistens nicht genau an seiner Summe in der HDT angepasst werden. z. B. werden bei den Ergebnissen vom Experiment „Top 100.000"-Datentabelle mit „k=2" die Summen der beiden Cluster zwar ungefähr richtig nach der RH der Attributwerte „pass" und „fail" in der HDT (PASS: 99 %, FAIL: 1 %) eingeteilt, jedoch werden die meisten FAIL-Arbeitsabläufe Cluster 0 zugeteilt. In diesem Fall kann Cluster 1 den Attributwert „fail" nicht repräsentieren. Die Cluster, die die bestimmten Attributwerte gut repräsentieren können, werden in Abschnitt 4.3 ausgewählt und interpretiert.

4.3 Weiterverarbeitung der Data Mining-Ergebnisse

Nach dem Vorgehensmodell von MESC von [ITP16] sollen folgende Aufgaben bearbeitet werden.

Tabelle 4.5: Aufgabendefinition der Weiterverarbeitung der Clusteranalyse-Ergebnisse (nach [ITP16])

6. Weiterverarbeitung der Data Mining-Ergebnisse	6.1 Extraktion handlungsrelevanter Data Mining-Ergebnisse	Unter Berücksichtigung der Handlungsrelevanz sowie technischen Maßzahlen sind für das SCM interessante Ergebnisse zu extrahieren
	6.2 Darstellungstransformation der Data Mining-Ergebnisse	In Abhängigkeit der eingesetzten Data Mining-Verfahren sowie der Aufgabenstellung müssen die Ergebnisse in eine explizite Darstellungsform überführt werden.

In diesem Abschnitt werden die DM-Ergebnisse weiterverarbeitet. Zuerst werden die handlungsrelevanten Data Mining-Ergebnisse mithilfe von Plot-Grafiken und Balkendiagrammen schriftlich beschrieben. Danach werden die erhaltenen Ergebnisse in eine explizite Darstellungsform überführt. Um die unterschiedlichen Cluster in den Grafiken besser identifizieren zu können und die Qualität der Abbildungen zu gewährleisten, werden alle relevanten Grafiken exportiert und im pdf-Format sowie in der originalen Größe auf der beigelegten CD gespeichert.

4.3.1 Extraktion handlungsrelevanter Clusteranalyse-Ergebnisse

Die handlungsrelevanten Clusteranalyse-Ergebnisse werden von der HDT mit unterschiedlichen Stichproben extrahiert, nämlich „Top 100.000"-HDT, „Last 100.000"-HDT, FDT, HRDT und More7RDT.

Die Interpretation der Ergebnisse wird hauptsächlich mithilfe der Statistik der Summe von den Datensätzen jedes Clusters, Plot-Grafik, Balkendiagrammen und den Experiment- ergebnissen, die in Tabelle 4.4 stehen, durchgeführt. Weil der Datentyp der Firmendaten nominal ist, bezieht sich der Interpretationsprozess hauptsächlich auf die Frage der „relativen Häufigkeit" (RH). Mithilfe eines Clusters werden unterschiedliche Attributwerte gruppiert und solche Gruppen von Attributwerten sind genau das Ziel der Clusteranalyse.

Der Interpretationsprozess wird in nachfolgenden Schritten durchgeführt. Zuerst werden die Experimentergebnisse von Abschnitt 4.2. interpretiert und analysiert. Durch diesen Schritt werden die nützlichen Clusteranalyse-Ergebnisse, die einen bestimmten Attributwert repräsentieren können, ausgewählt und im nächsten Schritt genau behandelt. Im zweiten Schritt werden die im ersten Schritt ausgewählten Ergebnisse mithilfe der Plot-Grafik oder der Balkendiagrammen und die Summe der Datensätze jedes Clusters genau analysiert und erläutert. Damit werden die versteckten Cluster unter den vorher festgelegten Analyseansätze herausgefunden.

Durch die Analyse der Tabelle 4.4 wird herausgefunden, dass viele durch die Clusteranalyse identifizierten Cluster die Attributwerte, die den vorher festgelegten Analyseansätze entsprechen, nicht genau repräsentieren können. Die Beurteilungskriterien über die Frage, ob ein Cluster einen bestimmten Attributwert repräsentieren kann, enthalten folgende zwei Aspekte:
1. Die Summe der Datensätze des Clusters soll ungefähr identisch zur Summe der Datensätze vom Attributwert sein, die den vorher festgelegten Analyseansätze entsprechen.
2. Die Summe der falsch klassifizierten Datensätze soll gering sein.
Nach der vorherigen Festlegung in Abschnitt 3.4 werden insgesamt zwei Analyseansätze für die Clusteranalyse festgelegt, nämlich „TotalResult" und „NmbOfRepairs" und „Häufigkeit". Für der Analyseansatz „TotalResult" sollen zwei Cluster gefunden werden, wobei ein Cluster die

PASS-Arbeitsabläufe und das andere Cluster die FAIL-Arbeitsabläufe repräsentieren soll. Für der Analyseansatz „NmbOfRepairs“ sollen fünf Cluster gefunden werden und jedes Cluster soll die folgenden Attributwerte repräsentieren: „no repair“, „low repair“, „middle repair“, „high repair“ und „repair>7“. Beim Analyseansatz „Häufigkeit“ sollen drei Cluster gefunden werden und jedes Cluster soll folgende Attributwerte repräsentieren: „low frequency“, „middle frequency“ und „high frequency“.

Nachfolgend werden nun die Ergebnisse nach zwei Clusteralgorithmen hintereinander interpretiert.

Ergebnisse des k-Means-Algorithmus

Identisch zur vorherigen Erläuterung werden die Parameter „max run“ und „max optimization steps“ mithilfe des „Centroid table“ festgelegt. Die genaue Anzahl der Parameter von unterschiedlichen k-Werten und Datentabellen wurde schon in Abschnitt 4.2.1 aufgelistet. Nach Tabelle im Anhang 18 ist die Höhe des Davies-Bouldin-Index in allen Experimenten ungefähr gleich. Es gibt keine extrem hohen oder extrem geringen Werte. Das heißt, dass die Qualität von allen Clustern ungefähr gleich ist. Der Analyseschwerpunkt liegt bei den Ergebnissen in Tabelle 4.4. Durch Analyse dieser Tabelle und die oben gezeigten zwei Beurteilungskriterien werden folgende Cluster ausgewählt:

Cluster 1 aus dem Experiment „Top 100.000“-Datenzeilen der HDT mit „k=2“, Cluster 4 des Experiments „Last 100.000“-Datenzeilen der HDT mit „k=5“, Cluster 2 des Experiments „FAIL“-Datentabelle mit „k=4“ und alle Experimente aus allen Datentabellen mit „k=3“.

Nach der Analyse der Ergebnisse aus Abschnitt 4.2 beginnt nun der zweite Schritt der Ergebnisinterpretation. Wegen der Größe der Plot-Grafik und zur Gewährleistung der Abbildungsqualität werden die relevanten Grafiken weder hier noch im Anhang gezeigt, sondern direkt exportiert und auf der beigelegten CD gespeichert. Im weiteren Verlauf werden die oben ausgewählten Cluster interpretiert.

1. Cluster 1 vom Experiment „Top 100.000“-Datenzeilen der HDT mit „k=2“: Dieses Cluster kann der Analyseansatz „TotalResult=PASS“ gut repräsentieren. Nach der Tabelle 4.4 sind alle Arbeitsabläufe dieses Clusters am Ende der Produktion bestanden. Durch die Analyse der Plot-Grafik werden folgende Ergebnisse erworben: Die Verteilung der bestehenden Arbeitsabläufe ist fast identisch zu der Verteilung der RH von allen Attributwerten in der HDT. Eine Ausnahme ist das Attribut „ManufacturingTime (Second) (ManuTime)_hFre.“. Die RH des Clusters 1 bei diesem Attributwert ist höher als die von diesem Attributwert in der HDT. Deshalb wird das Ergebnis erhalten, dass die Arbeitsabläufe, deren Produktionszeit zwischen 5 und 15 min beträgt, relative höhere PASS-Wahrscheinlichkeit als andere Arbeitsabläufe haben. Aus der Analyse lassen sich die folgenden Kenntnisse gewinnen: Die Verteilung von bestehenden Arbeitsabläufen in unterschiedlichen Attributwerten ist identisch zu deren Verteilung der RH in der HDT.

2. Cluster 4 von Experiment „Last 100.000“-Datenzeilen der HDT mit „K=5“: Dieses Cluster kann der Analyseansatz „NmbOfRepairs=low repair“ gut repräsentieren. Die Hälfte der Arbeitsabläufe dieses Clusters befindet sich in der Linie 3. Beim Attribut „ParaDesId“ befindet sich die Arbeitsabläufe dieses Cluster deutlich seltener bei den Id-Nummern 137, 164, 179. Die Verteilung der RH bei anderen Id-Nummern ist durchschnittlich. Beim Attribut „ProductId“ befinden die Arbeitsabläufe dieses Clusters sich durchschnittlich bei Produkt 6, 9, 10 und 12. Beim Attribut „WS“ befinden sich ungefähr 50 % der Arbeitsabläufe dieses Clusters beim

Attribut „WS_lFre.“, aber nur ungefähr 38 % beim Attribut „ WS_hFre.“. Die Verteilung der RH der Arbeitsläufe dieses Clusters in den Attributen „RouSe“, „WPId“ und „RS“ ist identisch zur Verteilung der RH von diesen Attributen in der HDT. Beim Attribut „ManuTime“ finden sich die Arbeitsabläufe jedoch hauptsächlich beim Attribut mit geringer Häufigkeit. Durch die Analyse werden folgende Ergebnisse erhalten: Die Arbeitsabläufe mit einer Reparaturhäufigkeit von 1 und 2 befinden sich besonders oft in der Linie 3 und wenn die Produktionszeit eines Arbeitsablaufs mehr als 30 min dauert, hat dieser Arbeitsablauf hohe Wahrscheinlichkeit, dass die Reparaturhäufigkeit 1 oder 2 beträgt. Die Arbeitsabläufe, die den „ParaDesId“ mit den Nummern 137, 164 und 179 zugeordnet sind, haben relativ geringe Wahrscheinlichkeit, dass ihre Reparaturhäufigkeit 1 oder 2 beträgt.

3. Cluster 2 des Experiments der FDT mit „k=4“: Dieses Cluster kann der Analyseansatz „NmbOfRepairs=middle repair“ gut repräsentieren. Die RH der Arbeitsabläufe dieses Clusters bei der Linie 4 ist deutlich kleiner als die in den anderen zwei Linien, wobei die RH der Arbeitsabläufe dieses Clusters ungefähr gleich sind. Beim Attribut „ParaDesId“ finden sich die Arbeitsabläufe dieses Clusters besonders oft beim Attribut „ParaDesId_24-70“. Bei den Attributen „ProductId“, „WS“, „RouSe“, „WPId“ und „RS“ finden sich die Arbeitsabläufe dieses Clusters hauptsächlich bei den jeweiligen Attributen mit hoher Häufigkeit. Unter Berücksichtigung der RH jedes Attributes in der HDT haben die Arbeitsabläufe bei den Attributen „Product_hFre.“, „WS_lFre.“, „RouSe_hFre.“ und „WPId_hFre.“ relativ hohe Wahrscheinlichkeit, dass ihre Reparaturhäufigkeit 3 oder 4 beträgt. Beim Attribut „ManuTime“ finden sich die Arbeitsabläufe dieses Clusters nur beim Attribut mit geringer Häufigkeit, das heißt, dass die Produktionszeit von allen Arbeitsabläufen, deren Reparaturhäufigkeit 3 oder 4 beträgt, über 30 min ist. Durch die Analyse können die Arbeitsabläufe von den Attributen, die hohe Wahrscheinlichkeit haben, dass ihre Reparaturhäufigkeit 3 oder 4 beträgt, festgestellt werden.

4. Drei Cluster von den Experimenten mit „k=3“: Durch die Analyse dieser drei Cluster im Experiment in der jeweiligen Datentabelle wird herausgefunden, dass diese Cluster nicht genau nach drei Häufigkeitsausmaßen definiert werden können. Deshalb wird dieses Ergebnis nicht weiter interpretiert, weil es der vorher festgelegten Analyseansatz nicht entsprechen kann.

Ergebnisse des EM-Algorithmus

Die Ergebnisse vom EM-Algorithmus werden nach identischen Verfahren zum k-Means-Algorithmus interpretiert. Zuerst werden die nützlichen Cluster aus Tabelle 4.4 ausgewählt nach den zwei Beurteilungskriterien, die im letzten Abschnitt schon vorgestellt wurden. Danach werden die ausgewählten Cluster interpretiert.

Nach Tabelle 4.4 und zwei Beurteilungskriterien zur Auswahl der Cluster, die einen bestimmten Attributwerten repräsentieren können, werden folgende Cluster ausgewählt: Cluster 0 vom Experiment „Last 100.000“-Datenzeilen der HDT mit „k=2“, Cluster 1 vom Experiment „Last 100.000“-Datenzeilen der HDT mit „k=5“ und Cluster 1 vom Experiment „FAIL“-Datentabelle mit „k=4“.

Nach dem ersten Clusterauswahlschritt beginnt nun der zweite Interpretationsschritt. Die Interpretation wird mithilfe der Balkendiagramme durchgeführt. Alle relevanten Balkendiagramme werden zur Gewährleistung der Qualität der Abbildungen exportiert und auf der begleitenden CD gespeichert.

1. Cluster 0 vom Experiment „Last 100.000“-Datenzeilen der HDT mit „k=2“: Dieses Cluster kann der Analyseansatz „TotalResult=PASS“ gut repräsentieren. Durch die Analyse der Balkendiagramme können die folgenden Ergebnisse erhalten werden: Die Arbeitsabläufe dieses Clusters haben hauptsächlich das Attribut „no repair“ und ungefähr 80 % von allen Arbeitsabläufen, die dem Attribut „low repair“ zugeordnet werden, gehören zum Cluster 0. Die Verteilung der Arbeitsabläufe bei drei Linien ist identisch in ihrer RH-Verteilung zur HDT. Beim Attribut „ProcessId“ enthält das Cluster 0 die kompletten Arbeitsabläufe, die dem Prozess 99 zugeordnet sind, und die meisten Arbeitsabläufe, die dem Prozess 1 zugeordnet sind. Beim Attribut „ParaDesId“ entspricht die Verteilung der Arbeitsabläufe von Cluster 0 der RH-Verteilung jedes „ParaDesId“ in der HDT. Beim Attribut „ProductId“ enthält das Cluster 0 die kompletten Arbeitsabläufe bei den Produkten 11 und 14 (jedoch gehören die beiden ProduktId zum Attributwert „geringe Häufigkeit“ beim k-Means-Algorithmus). Die Verteilung der Arbeitsabläufe in anderen Produkten entspricht in ihrer RH-Verteilung der HDT. Bei den Attributen „RS“, „RouSe“, „WPId“, „WS“ und „Remarks“ ist die Verteilung der jeweiligen Arbeitsabläufe des Clusters 0 identisch zu ihrer Verteilung beim Attribut „ProductId“. Das heißt: Das Cluster 0 enthält die kompletten Arbeitsabläufe von den Attributwerten „geringer Häufigkeit“ beim k-Means-Algorithmus und die Verteilung der Arbeitsabläufe des Clusters 0 von den übrigen Attributwerten entspricht der Verteilung der RH des jeweiligen Attributwertes in der HDT. Beim Attribut „ManuTime“ enthält Cluster 0 jedoch die kompletten Arbeitsabläufe von den Attributwerten „hoher Häufigkeit“ beim k-Means-Algorithmus. Bei anderen Attributwerten entspricht die Verteilung der Arbeitsabläufe dieses Clusters der RH-Verteilung jedes Attributwertes in der HDT. Durch diese Analyse wird ermittelt, dass die Verteilung der Arbeitsabläufe des Clusters 0 in jedem Attributwert seiner RH-Verteilung in der HDT ungefähr entspricht. Dieses Ergebnis ist identisch zum Ergebnis mit dem k-Means-Algorithmus.
2. Cluster 1 vom Experiment „Last 100.000“-Datenzeilen der HDT mit „K=5“: Dieses Cluster kann der Analyseansatz „NmbOfRepairs=no repair“ gut repräsentieren. Beim Attribut „ProcessId“ befinden die Arbeitsläufe vom Cluster 1 nur beim Prozess 1. Bei den Attributen „LineId“, „ProductId“ und „RS“ entspricht die Verteilung der Arbeitsabläufe dieses Clusters genau der RH-Verteilung jedes Attributwertes in der HDT. Beim Attribut „ManuTime“ ist die Verteilung der Arbeitsabläufe dieses Clusters etwas geringer als das andere Cluster bei den Attributwerten, deren Produktionszeit mehr als 30 min beträgt. Beim Attribut „ParaDesId“ belegen die Arbeitsabläufe dieses Clusters eine viel geringere RH bei den „ParaDesId“ mit der Nummern 137, 164, 179, die die höchsten RH in der HDT belegen. Bei „ParaDesId“ 27 gehören die Arbeitsabläufe komplett zu einem anderen Cluster. Beim Attribut „RouSe“ finden sich die Arbeitsabläufe dieses Cluster nicht bei den Attributwerten „RouSe“ 10 und 115 und die RH dieses Cluster beim Attributwert „RouSe“ 80 ist relativ höher als die von anderen „RouSe“-Nummern. Die Verteilung der Arbeitsabläufe dieses Clusters in anderen „RouSe“-Nummern entspricht der Verteilung der RH jeder „RouSe“-Nummer in der HDT. Beim Attribut „WPId“ gehören die Arbeitsabläufe des „WP 80“ zu anderen Clustern. Die Verteilung der Arbeitsabläufe dieses Clusters in anderen „WPId“-Nummern entspricht der Verteilung der RH jeder „WPId“-Nummer in der HDT. Beim Attribut „WS“ gehören keine Arbeitsabläufe dieses Clusters zu den „WS“ 1, 5 und 10-31“. Die RH der Arbeitsabläufe dieses Clusters bei den „WS“ 2, 3, 4, 9 ist relativ geringer als die RH bei anderen „WS-Nummern“. Beim Attribut „Remarks“ gehören keine Arbeitsabläufe dieses Clus-

ters zu den Attributwerten „AESStart: PASS“ und die meisten „Remarks“, die mit dem Zeichen „FAIL“ markiert werden. Die Verteilung der Arbeitsabläufe dieses Clusters bei den anderen Attributwerten entspricht der RH-Verteilung dieser Attributwerte in der HDT. Beim Attribut „ManuTime“ zeigen die Arbeitsabläufe dieses Clusters eine relativ hohe RH bei den Attributwerten mit einer Produktionszeit von über 30 min. Durch die Analyse werden die Kenntnisse erworben, dass bei manchen Attributwerten relativ höhere Reparaturwahrscheinlichkeit als bei anderen Attributwerten bestehen und bei manchen Attributwerten Reparaturen zu 100 % erfolgen werden.

3. Cluster 1 vom Experiment „FAIL-Datentabelle“ mit „k=4“: Dieses Cluster kann der Analyseansatz „NmbOfRepairs=low repair“ gut repräsentieren. Beim Attribut „ProcessId“ gehören die kompletten Arbeitsabläufe von Prozess 99 und die meisten Arbeitsabläufe von Prozess 1 zu diesem Cluster. Bei den Attributen „LineId“, „ProductId“ entspricht die Verteilung der Arbeitsabläufe dieses Clusters der Verteilung der RH jeder „LineId“ in der FDT. Beim Attribut „ParaDesId“ gehören die Arbeitsabläufe von fast allen „ParaDesId“ zu diesem Cluster. Die meisten Arbeitsabläufe von den „ParaDesId“ mit den Nummern „42, 43, 51, 91, 98, 143, 148“ gehören zu einem anderen Cluster. Bei den Attributen „WS“, „RS“, „RouSe“, „WPId“ und „Remarks“ ist die Verteilung der Arbeitsabläufe identisch zu der vom Attribut „ParaDesId“. Das heißt: Die Arbeitsabläufe der meisten Attributwerte gehören zu diesem Cluster und die Arbeitsabläufe anderer Cluster befinden sich meist bei den Attributwerten, die hohe RH bei der FDT haben. Beim Attribut „ManuTime“ entspricht die Verteilung der Arbeitsabläufe dieses Clusters der RH-Verteilung jedes Attributwertes in der FDT, dessen Produktionszeit mehr als 30 min beträgt. Durch die Clusteranalyse werden die folgenden Kenntnisse erworben: Die Arbeitsabläufe, die am Ende durchgefallen sind, haben ihre Reparaturhäufigkeit hohe Wahrscheinlichkeit, zwischen 1 und 2 zu liegen. Die Reparaturhäufigkeit der Arbeitsabläufe von allen Attributwerten mit mittlerer und geringer Häufigkeit vom jeweiligen Attribut beträgt meist zwischen 1 und 2.

4. Durch die Ergebnisse vom k-Means-Algorithmus werden die Kenntnisse genommen, dass die drei Cluster bei den Experimenten mit „K=3“ nicht nach drei Häufigkeitsausmaßen definiert werden können. Damit entsprechen die Ergebnisse nicht dem vorher festgelegten Analyseansatz. Deshalb werden solche Experimentergebnisse mit „k=3“ vom EM-Algorithmus nicht interpretiert.

4.3.2 Darstellungstransformation der Clusteranalyse-Ergebnisse

In Abschnitt 4.3.1 wurden 6 Cluster aus den Clusteranalyse-Ergebnissen von Abschnitt 4.2 ausgewählt und schriftlich interpretiert. Diese Interpretationsform ist nicht übersichtlich. In dieser Masterarbeit werden die schriftlichen Clusteranalyse-Ergebnisse zu zwei Darstellungsformen transformiert, um die genauen Informationen von den Clustern übersichtlich und explizit darzustellen, nämlich die Tabellenform und die Grafikform.

Tabellenform

Nachfolgend werden zwei Tabellen zur Darstellung der mit den unterschiedlichen Experimenten erhaltenen Clusteranalyse-Ergebnisse erstellt. Wegen der Größe werden die beiden Tabellen in den Anhang 20 und Anhang 21 angezeigt. Die beiden Darstellungstabellen werden nach folgenden Gedanken aufgebaut: Die Tabelle im Anhang 20 besteht aus drei Clustern, die durch den

k-Means-Algorithmus gefunden wurden, und zeigt alle Attribute, die in der Clusteranalyse analysiert werden. Die RH jedes Attributwertes wird direkt hinter dessen zugehörigem Attribut in der Tabelle aufgeführt, um den Verteilungszustand der Arbeitsabläufe dieses Clusters zwischen unterschiedlichen Attributwerten innerhalb eines Attributes zu beschreiben. Die RH wird als Prozentzahl ausgedrückt. Danach werden die RH jedes Attributwertes im Cluster mit ihren RH in der HDT verglichen, die in Abschnitt 3.4 schon aufgelistet sind. Wenn die beiden RH einen großen Unterschied zeigen, werden die RH dieses Attributwertes aus der HDT direkt dahinter in einer Klammer zusätzlich angegeben. Identisch zum Analyseverfahren in Abschnitt 3.4 soll, wenn die beiden Prozentzahlen eines Attributwertes stark unterschiedlich sind, dieser Attributwert besonders beachtet werden.

Die Tabelle im Anhang 21 wird nach einem ähnlichen Verfahren aufgebaut. Weil der Datentyp der mit dem EM-Algorithmus analysierten Daten nominal ist, umfassen manche Attribute zahlreiche Attributwerte, z. B. das Attribut „ResultSequence“. Deshalb werden nicht alle Attributwerte von solchen Attributen mit deren RH in der Tabelle aufgelistet. Bei solchen Attributen wird die Zelle für dieses Attribut in der Tabelle in zwei Abschnitte unterteilt. Die Attributwerte, deren RH im Cluster fast identisch zu seiner RH in der HDT sind, werden extra in der unteren Zelle ohne Prozentzahlen aufgeführt, damit die komplette Verteilung der RH jedes Attributwertes des Clusters angezeigt werden kann.

Grafikform

Die Clusteranalyse-Ergebnisse werden mithilfe von Plot-Grafiken und Balkendiagrammen visuell interpretiert. Alle relevanten Grafiken werden exportiert und auf der beigelegten CD gespeichert.

Nun wird das Verfahren zum Lesen der Informationen aus den Grafiken kurz vorgestellt. Bei der Plot-Grafik stehen die Attributwerte, die zugehörigen Attribute und gegebenenfalls die RH von manchen Attributwerten auf der horizontalen Koordinatenachse. In der vertikalen Koordinatenachse stehen die RH eines Clusters bei jedem Attributwert von einem Attribut, z. B. beträgt die RH vom Cluster 0 bei drei „LineId“ jeweils 30 %, 50 % und 20 % und diese drei Prozentzahlen können mithilfe der vertikalen Koordinatenachse abgerufen werden.

In dieser Masterarbeit sind die Balkendiagramme zur Interpretation der Clusteranalyse-Ergebnisse vom EM-Algorithmus die gestapelten Balkendiagramme. Die vertikale Koordinatenachse zeigt die Namen der Attributwerte. Die horizontale Koordinatenachse zeigt die Häufigkeit jedes Attributwertes. Die Häufigkeit der gestapelten Attribute wird durch die folgende Formel ermittelt:

$$Häufigkeit\ des\ gestapelten\ Attributes = Prozentzahl\ der\ gestapelten\ Attribute\ im\ Balken * Häufigkeit\ dieses\ Balkens \quad \textbf{(Formel 4.1)}$$

4.4 Fazit

In diesem Kapitel wurde das Clusterverfahren auf die Firmendaten angewendet, um versteckte Cluster herauszufinden. Weil die meisten Clusteralgorithmen nur für numerische Daten geeignet sind und viele Datentypen bei den Firmendaten nominal sind, wird zuerst der „fachliche Kodierungs“-Prozess durchgeführt, damit die Clusteralgorithmen für die numerischen Daten auch auf

die Firmendaten angewendet werden können. Bei diesem Prozess wird die originale Datentabelle erweitert. Der Prozess der „fachlichen Kodierung“ produziert zahlreiche neue Attribute. Um das Dataset zu komprimieren, wird die Anzahl der „Relativen Häufigkeit“ als Klassifikationskriterium der Aggregationsstufe gewählt und die neu produzierten Attribute werden nach ihrer RH im originalen Attribut in drei neuen Attributen aggregiert, nämlich „high frequency“, „middle frequency“ und „low frequency“, damit die Attribute, die aggregiert werden, unter Berücksichtigung ihrer RH im originalen Attribut homogen sind. Nach der Vorbereitung der Clusteranalyse wurden zwei ausgewählte Clusteralgorithmen durchgeführt und die Ergebnisse validiert. Im letzten Abschnitt wurden die Ergebnisse in drei Formen interpretiert, nämlich in schriftlicher Form, Tabellenform und Grafikform.

Weil das Clusterverfahren ein „unsupervised“ DM-Verfahren ist, werden die Cluster automatisch herausgefunden. Das heißt, dass das Clusterverfahren das Cluster nicht genau nach dem Bedarf herausfinden kann. In dieser Masterarbeit werden vier Cluster nach den Analyseansätzen durch zahlreihe Experimente gefunden, nämlich Cluster_pass, Cluster_no repair, Cluster_low repair und Cluster_middle repair. Weil die RH des Attributwertes „fail“ im Attribut „TotalResult“ und die RH der Attributwerte „high repair“ und „repair>7“ im Attribut „NmbOfRepairs“ gering sind, sind die Cluster von den drei Attributwerten schwer zu finden. In Abschnitt 3.4 wurden die Attributwerte, die hohe Wahrscheinlichkeiten von „fail“, „high repair“ und „repair>7“ haben, durch Vergleichstabellen in unterschiedlichen Attributen herausgefunden und die Ergebnisse dieses Abschnitts dienen als Ergänzung der Clusteranalyse. Mithilfe der Ergebnisse der Datenvorverarbeitung und der Clusteranalyse werden die versteckten Gruppen von allen Attributwerten der beiden Analyseansätze herausgefunden. Nun werden die Ergebnisse der Clusteranalyse zusammengefasst.

Durch die Analyse der Tabellen 4.3.2 und 4.3.3 zeigt sich: Die Verteilung der Arbeitsabläufe im Cluster_pass ist fast identisch zu ihren Verteilung in der HDT. Die Unterschiede der RH der Arbeitsabläufe finden sich hauptsächlich bei den Attributen „NmbOfRepairs“, „WorkSequence“, „ManufacturingTime(Second)“ und „Remarks“. Wenn ein Arbeitsablauf während des Produktionsprozesses keine Reparatur durchlaufen hat, sich auf die WS 2 bis 8 außer WS 5 befindet, und seine Produktionszeit zwischen 5 und 15 min beträgt, hat er hohe Wahrscheinlichkeit, dass er am Ende den Qualitätstest besteht. Wenn der Arbeitsablauf am Ende besteht, hat er nur geringe Wahrscheinlichkeit, dass das Attribut „Remarks“ mit „FAIL“ markiert wird.

Die Verteilung der RH der Arbeitsabläufe vom Cluster „no repair“ ist nicht identisch zu ihrer Verteilung in der HDT. Die Unterschiede in der RH finden sich hauptsächlich in den Attributen „ProcessId“, „ParaDesId“, „WS“, „RouSe“, „WPId“ und „ManuTime“. Wenn ein Arbeitsablauf keine Reparatur während der Produktion hat, befindet er sich wahrscheinlich in bestimmten Attributwerten von den obergenannten Attributen, die bei der Interpretation der Clusteranalyse-Ergebnisse vom Cluser_no repair schon genau erläutert werden. Es ist besonders zu beachten, dass die Produktionszeit der Arbeitsabläufe ohne Reparatur eine geringere Wahrscheinlichkeit hat, mehr als eine Stunde zu dauern. Weiterhin sind die „Remarks“ von Arbeitsabläufen, die bis zum Ende laufen, mit „PASS“ markiert. Das heißt, dass „PASS“ im Attribut „Remarks“ Zusammenhänge mit dem endgültigen Ergebnis „PASS“ haben.

Die Verteilung der RH der Arbeitsabläufe beim Cluster_low repairs“ ist unterschiedlich im Vergleich zu den letzten zwei Clustern. In dieser Masterarbeit werden zwei Cluster vom Attri-

butwert „low repair“ separat durch zwei Clusteralgorithmen herausgefunden. Weil das Cluster, das durch den k-Means-Algorithmus herausgefunden wird, relativ hohe RH von Arbeitsabläufen mit „middle repair“, „high repair“ und „repair>7“ enthält, ist die genaue RH-Verteilung der Arbeitsabläufe von beiden Clustern unterschiedlich.

Nun werden die gemeinsamen Merkmale von beiden Clustern erläutert. Die Verteilung der Arbeitsabläufe von den Attributen „NmbOfRepairs“, „WS“, „RS“ und „ManuTime“ in beiden Clustern ist unterschiedlich im Vergleich zu den vorherigen zwei Clustern. Die beiden Cluster enthalten keinen Attributwert „no repair“. Beim Attribut „WS“ wird die RH der Attributwerte mit hoher Häufigkeit in der HDT deutlich reduziert und gleichzeitig wird die RH der Attributwerte mit geringer Häufigkeit stark erhöht. Die Produktionszeit der Arbeitsabläufe, deren Reparaturanzahl zwischen 1 und 2 beträgt, ist meistens höher als eine Stunde. Durch das Ergebnis des Attributes „Remarks“ des EM-Algorithmus wird herausgefunden, dass die Summe der „Remarks“, die mit „FAIL“ markiert werden, und ihre RH deutlich höher sind als die im Cluster_„no repair“. Das heißt: Die Erhöhung der Reparaturhäufigkeit eines Arbeitsablaufs erhöht gleichzeitig auch die Wahrscheinlichkeit des Durchfallens von den einzelner Operationen, die zu diesem Arbeitsablauf gehören.

Durch die Analyse der Verteilung der RH beim Cluster_middle repair wird herausgefunden, dass die Verteilung der RH jedes Attributwertes in diesem Fall stark unterschiedlich zu den letzten drei Fällen ist. Die genauen RH wurden in der Tabelle 4.3.2 ausführlich aufgeführt und nicht wiederholt.

Nach der obigen Analyse von vier Clustern wird zusammengefasst, dass die Verteilung der RH der Attributwerte von den Attributen „WorkSequence“ und „ManufacturingTime(Second)“ besonders von der Häufigkeit der Reparaturen beeinflusst werden kann. Je höher die Reparaturhäufigkeit ist, desto größere Unterschiede der RH jedes Attributwertes im Vergleich mit ihren RH in der HDT treten auf. Die Verteilung der RH jedes Attributwertes von den Arbeitsabläufen, die am Ende bestanden haben, ist fast identisch zu jener in der HDT.

5. Praktische Verwertbarkeit des Vorgehensmodells

Grundsätzlich ist die Struktur dieser Masterarbeit mithilfe des Vorgehensmodells zur Musterextraktion in SCs (MESC) aufgebaut, das in Abschnitt 2.2.1 schon vorgestellt wurde. Der Modellierungsprozess dieser Masterarbeit wird genau nach den Schritten dieses Vorgehensmodells durchgeführt. In diesem Kapitel wird nun das Vorgehensmodell zur MESC bezogen auf seine Verwertbarkeit anhand der konkreten Durchführung der Datenvorverarbeitung und Clusteranalyse in dieser Masterarbeit betrachtet.

Weil die Schwerpunkte dieser Masterarbeit die Datenvorverarbeitung und die Clusteranalyse sind, wird sich die Bewertung des Vorgehensmodells auf die Phasen „Datenaufbereitung", „Vorbereitung des Data Mining-Verfahrens", „Anwendung des Data Mining-Verfahrens" und „Weiterverarbeitung der Data Mining-Ergebnisse" konzentrieren. Die dritte Phase des Vorgehensmodells, „Datenaufbereitung", enthält vier Schritte, nämlich „Format-Standardisierung", „Gruppierung", „Datenanreicherung" und „Transformation". Durch die praktische Durchführung des Datenvorverarbeitungsprozesses in RapidMiner wird herausgefunden, dass sich die meisten Schritte des Datenvorverarbeitungsprozesses auf den vierten Schritt „ Transformation" konzentrieren. Das heißt, dass dieser Schritt weiter spezifiziert werden kann. In dieser Masterarbeit wird das Vorgehensmodell dieser Phase auf Grundlage des Vorgehensmodells zur MESC und seiner Verwertbarkeit in RapidMiner modifiziert und optimiert. Der Datenvorverarbeitungsprozess wird hier durch drei Methoden durchgeführt und alle Schritte, die im Vorgehensmodell zur MESC stehen, werden diesen drei Methoden zugeordnet, nämlich Datenhomogenisierung, Datenaggregation und Feature Selection. Nachfolgend wird das modifizierte Vorgehensmodell für die Datenvorverarbeitung mithilfe der Tabelle 5.1 dargestellt und dieses modifizierte Vorgehensmodell für die Datenvorverarbeitung wird als wissenschaftlicher Beitrag dieser Masterarbeit geleistet.

Tabelle 5.1: **Modifiziertes Vorgehensmodell zur Datenvorverarbeitung**

Phase	Methode	Schritte	Aufgaben
3. Datenvorverarbeitung	Datenaggregation	Datenintegration	Fachliche Gruppierung von Datasets
			Redundanzanalyse nach der Gruppierung
		Datenanreicherung	Attributextraktion und -konstruktion
			Diskretisierung
			Import von externen Attributen
	Datenhomogenisierung	Prüfung der Voraussetzung für die Datenvorverarbeitung	Prüfung auf Format-Standardisierung der Datasets
			Prüfung auf Atomarität der Attribute
		Bereinigung der fehlenden Werte	Technische Kodierung der fehlenden Werte und einheitliche Filterung
			Manuelle Festlegung und direkte Eingabe
		Bereinigung der verrauschten Daten	Diskretisierung

			Technische Kodierung der verrauschten Daten und einheitliche Filterung
		Datentyptransformation	Direkte Datentyptransformation nach dem Bedarf der Datenanalyse durch Operatoren in RapidMiner
			Diskretisierung
	Feature Selection	Manuelle Auswahl	Bereinigung der Redundanzattribute
		Auswahl durch Algorithmen	Auswahl eines geeigneten Feature Selection-Algorithmus nach dem Bedarf der Datenanalyse und dem Datentyp der Daten im Datasets

In Tabelle 5.1 werden die Schritte der Datenvorverarbeitung in drei Methoden zusammengefasst. Nach den praktischen Erfahrungen der Modellierung im RapidMiner werden einige neue Schritte addiert und diesen drei Methoden zugeordnet. In der rechten Tabellenspalte zu jedem Schritt werden die Aufgaben aufgelistet, die in dieser Masterarbeit zur Durchführung dieses Schrittes behandelt wurden.

In dieser Masterarbeit wird das Clusterverfahren nach der Fragestellung als DM-Verfahren auf die Firmendaten angewendet. Der Modellierungsprozess in Kapitel 4 wird nach den Schritten dieses Vorgehensmodells aufgebaut. Alle Schritte passen genau zum Modellierungsprozess des Clusterverfahrens gemäß dieser Masterarbeit. Zusätzlich wird nach dem praktischen Bedarf in dieser Masterarbeit ein Schritt „Clustervalidierung“ in der fünften Phase als dritter Punkt eingeführt, um die besten Ergebnisse aus dem Clusterverfahren auszuwählen, damit die besten DM-Ergebnisse in der Phase 6 weiterverarbeitet und interpretiert werden können.

6. Zusammenfassung und Ausblick

In dieser Masterarbeit wurde das Data Mining-Verfahren auf produktionslogistische Massendaten angewendet, die von einer Firma bereitgestellt wurden. Der Schwerpunkt dieser Masterarbeit war die Datenvorverarbeitung und die spezifische Fragestellung dieser Masterarbeit das Herausfinden von Clustern aus diesen Firmendaten mithilfe des Clusterverfahrens. In Kapitel 2 wurden die notwendigen theoretischen Kenntnisse über die Methoden der Datenvorverarbeitung, Methoden der Ähnlichkeitsmaßen, Clusteralgorithmen vorgestellt. In Kapitel 3 wurde der Modellierungsprozess vorbereitet und drei Datenvorverarbeitungs-methoden wurden zur Vorverarbeitung für das DM-Verfahren auf die Firmendaten angewendet, nämlich Datenhomogenisierung, Datenaggregation und Feature Selection. Nach dieser Datenvorverarbeitung waren die Firmendaten geeignet, durch das DM-Verfahren analysiert zu werden. In Kapitel 4 wurden zwei Clusteralgorithmen zur Extraktion von versteckten Clustern innerhalb der Daten auf die Ergebnisse der Datenvorverarbeitung angewendet. Nach der Datenanalyse werden zwei Analyseansätze für die Firmendaten festgelegt, nämlich „TotalResult“ und „NmbOfRepairs“. Mithilfe des Clusterverfahrens wurden einige Cluster gefunden, die die Attributwerte der beiden Analyseansätze repräsentieren können. Grundsätzlich ist die Struktur dieser Masterarbeit nach dem Vorgehensmodell zur MESC aufgebaut, das vom Lehrstuhl für Produktion und Logistik bereitgestellt wird. In Kapitel 5 wurde die praktische Verwertbarkeit dieses Vorgehensmodells gezeigt.

Die erste Herausforderung dieser Masterarbeit war die anfängliche Datenanalyse der Firmendaten, weil die originalen Firmendaten ungeordnet vorlagen und komplex waren. Durch die anfängliche Datenanalyse sollte ein Zielformat für die relevanten Datenbestände der Firmendaten entwickelt werden. Das Ziel dieses Zielformats war, die Datenmenge ohne großen Datenverlust zusammenzuführen und zu komprimieren. Deshalb wurde ein ER-Modell zur Analyse der Zusammenhänge zwischen unterschiedlichen Datentabellen aufgebaut und es wird in der beigelegt CD gespeichert.

Die zweite Herausforderung dieser Masterarbeit bestand in dem Herausfinden der passenden Methoden und Algorithmen zur Durchführung des DM-Prozesses. Die meisten Daten aus den Firmendaten waren zwar Ziffern, die aber keine numerische Bedeutung hatten. Deshalb konnten die normalen Datenvorverarbeitungsmethoden, die nur für numerische Daten geeignet sind, nicht auf die Firmendaten angewendet werden. Dieses Problem betraf auch die Clusteranalyse. Deshalb sollten die Datentypen innerhalb der Firmendaten mithilfe von unterschiedlichen Methoden transformiert und die dafür geeigneten DM-Methoden gesucht werden.

Die dritte Herausforderung dieser Masterarbeit war die Festlegung des Analyseansatzes. Nach der Datenvorverarbeitung wurden die originalen Daten komprimiert und geordnet. Die Eigenschaft jedes Datensatzes wurde mithilfe von unterschiedlichen Attributwerten festgestellt, aber die Bedeutung der meisten Attributwerte ist unbekannt. Zwar ist die Clusteranalyse ein „unsupervised“ DM-Verfahren, aber eine vorherige Festlegung des Analyseansatzes ist notwendig, um die Ergebnisse später besser interpretieren zu können. Nach der Datenanalyse wurden zwei Attribute als Analysesansätze festgelegt, nämlich „TotalResult“ und „NmbOfRepairs“, weil die beiden Attribute die wichtigsten Kriterien zur Beurteilung der Qualität eines Arbeitsablaufs innerhalb der Firmendaten sind.

Die vierte Herausforderung dieser Masterarbeit war die Herausnahme von Stichproben. Weil die Summe an Datensätzen der integrierten Haupttabelle 130 Mio. beträgt, mussten Stichproben aus dem Datenbestand gezogen werden. Um sich im DM-Prozess mehr auf den Analyseansatz konzentrieren zu können, wurden Stichproben zusätzlich entsprechend den Attributwerten „fail“, „high repair“ und „repair>7“ für die beiden Analyseansätze gebildet. Weil diese drei Attributwerte nur geringe relative Häufigkeiten in dem jeweiligem Attribut belegen, ist die Stichprobe viel genauer als die aus der Hauptdatentabelle mit einer gleichen Summe an Datensätzen.

Die fünfte Herausforderung dieser Masterarbeit war die Modellierung. Um die theoretischen Kenntnisse der Datenvorverarbeitung und Clusteranalyse auf die Firmendaten anwenden zu können, ist die Modellierung in Form geeigneter Modelle notwendig. Gleichzeitig ist die Festlegung der Parameter in RapidMiner eine schwierige Aufgabe im Modellierungsprozess, wobei sie nach dem Datenzustand und dem Datenanalysebedarf durch zahlreiche Experimente erfolgte.

Die sechste Herausforderung dieser Masterarbeit bestand in der Interpretation der Ergebnisse. Die Datenvorverarbeitungs- und Clusteranalyse-Ergebnisse in dieser Masterarbeit werden mithilfe von drei Formen interpretiert, nämlich in schriftlicher Form, Tabellenform und Grafikform.

Für die zukünftige Untersuchungsrichtung des Themas dieser Masterarbeit werden nun einige lohnenswerte Aufgaben als Ausblick aufgezählt. In Abschnitt 2.4.2 wurde das Clusterverfahren mit dem Klassifikationsverfahren verglichen. Das Cluster wird auch eine implizite Klasse genannt und das Clusterverfahren wird auch als automatisches Klassifikationsverfahren betrachtet. Eine wichtige Aufgabe dieser Masterarbeit ist, durch das Clusterverfahren einige nützliche Richtungen für die spätere Klassifikationsanalyse zu bieten. In dieser Arbeit werden die Cluster der Firmendaten entsprechend zwei Analyseansätze bzw. Attributen gesucht, nämlich „TotalResult“ und „NmbOfRepairs“. Die Arbeitsabläufe bzw. „Tag“-Nummern, deren Attributwerte „fail“, „high repair“ und „repair>7“ sind, werden als problematische Arbeitsabläufe betrachtet. Weil die Clusteranalyse ein „unsupervised“ Verfahren ist, können die Firmendaten nicht absolut nach eigenen Wünschen genau klassifiziert werden. Deshalb ist das Klassifikationsverfahren notwendig. Für die zukünftige Untersuchung ist eine Weiterarbeit der Klassifikation von den bisherigen Firmendaten nach den oben genannten beiden Analyseansätzen empfehlenswert. Nach der Durchführung des Klassifikationsverfahrens sollen dessen Ergebnisse mit den Ergebnissen der Clusteranalyse in dieser Masterarbeit verglichen werden, damit die Firmendaten nach beiden Analyseansätzen genau klassifiziert werden können und die problematischen Arbeitsabläufe und ihre zugehörigen Attributwerte besser identifiziert werden können.

In dieser Masterarbeit wurden vier Cluster durch die Clusteranalyse herausgefunden, die vier Attributwerte von beiden Analyseansätze repräsentieren können, nämlich „pass“, „no repair“, „low repair“ und „middle repair“. Außerdem können die Ergebnisse der Datenvorverarbeitung in Abschnitt 3.4 die Attributwerte „fail“, „high repair“ und „repair>7 von beiden Analyseansätze repräsentieren. Im späteren Klassifikationsprozess können die Richtungen aus den Ergebnissen von der Datenvorverarbeitung und Clusteranalyse festgelegt werden.

Durch die Datenanalyse wird herausgefunden, dass die Bedeutung vieler Attributwerte der Firmendaten unbekannt ist, z. B. „ResultSequence 1“. Es ist empfehlenswert, die genaue prakti-

sche Bedeutung von allen unbekannten Attributwerten in der Firma zu recherchieren, um die Klassifikationsergebnisse besser interpretieren zu können.

Insgesamt konnte im Rahmen der vorliegenden Arbeit jedoch gezeigt werden, dass sich mit Hilfe von Datenvorverarbeitung auch unübersichtlich große Datenbestände mit unterschiedlichsten Attributformen und -werten durch genaue Analyse der Daten und Clusteranalyse auf handhabbare Größen reduzieren lassen, aus denen relevante Informationen über den Produktionsprozess abgeleitet werden können.

Literaturverzeichnis

Bücher

[Agg15]: Aggarwal, Charu C.: *Data Mining – The Textbook*, New York, Springer, 2015. ISBN: 978-3-319-14142-8

[BKN08]: Blattberg, Robert C.; Kim, Byung-Do; Neslin, Scott A.: *Database Marketing – Analyzing and Managing Customers*, New York, Springer Science+Business Media Verlag, 2008. ISBN: 978-0-387-72579-6

[Blu06]: Blum, Ingo: *Data-Mining-Techniken im Marketing und Vertrieb – Grundlagen, Methoden und Funktionsweisen*, Saarbrücken, VDM Verlag, 2006. ISBN: 978-3-86550-0630-6

[Bra07]: Bramer, Max: *Principles of Data Mining*, London, Springer, 2007. ISBN: 978-1-84628-765-7

[BS13]: Bandyopadhyay, Sanghamitra; Saha, Sriparna: *Unsupervised Classification – Similarity Measures, Classical and Metaheuristic Approaches, and Applications*, Berlin Heidelberg, Springer Verlag, 2013. ISBN: 978-3-642-32451-2

[BSA15]: Bolón-Canedo, Verónica; Sánchez-Maroño, Noelia; Alonso-Betanzos, Amparo: *Feature Selection for High-Dimensional Data*, Heidelberg, Springer Verlag, 2015. ISBN: 978-3-319-21858-8

[BW73]: Berling, Gerd; Wersig, Gernot: *Zur Typologie von Daten und Informationssystemen – Terminologie, Begriffe u. Systematik*, Pullach, Dokumentation Verlag 193. ISBN: 3-7940-3406-9

[Cic15]: Cichosz, Pawel: *Data Mining Algorithms - Explained Using R*, West Sussex, Wiley Verlag, 2015. ISBN: 978-1-118-33258-0

[CL16]: Lämmel, Uwe; Cleve, Jürgen: *Data Mining*, 2. Aufl., Berlin, Oldenbourg Verlag, 2016. ISBN: 978-3-11-045677-6

[CPS+07]: Cios, Krzysztof J.; Pedrycz, Witold; Swiniarski, Roman W.; Kurgan, Lukas A.: *Data Mining – A Knowledge Discovery Approach*, New York, Springer Science+Business Media Verlag, 2007. ISBN: 978-0-387-36795-8

[ELL+11]: Everitt, Brian S; Landau, Sabine; Leese, Morven; Stahl, Daniel: *Cluster Analysis*, 5. Aufl.West Sussex, Wiley Verlag, 2011, ISBN: 978-0-470-74991-3

[FR06]: Faroult, Stephane; Robson, Peter: *The Art of SQL*, Sebastopol, O'Reilly Verlag, 2006. ISBN: 0-596-00894-5

[GA13]: Gressner, Axel M.; Arndt, Torsten: *Lexikon der Medizinischen Laboratoriumsdiagnostik*, 2. Aufl. Berlin Heidelberg, Springer Verlag, 2013. ISBN: 978-3-642-12921-6

[GLH15]: García, Salvador; Luengo, Julián; Herrera, Francisco: *Data Preprocessing in Data Mining*, Heidelberg, Springer Verlag, 2015. ISBN: 978-3-319-10247-4

[GMJ13]: Georgieva, Petia; Mihaylova, Lyudmila; Jain, Lakhmi C.: *Advances in Intelligent Signal Processing and Data Mining – Theory and Applications*, Berlin Heidelberg, Springer Verlag, 2013. ISBN: 978-3-642-28695-7

[GMW07]: Gan, Guojun; Ma, Chaoqun; Wu, Jianhong: *Data Clustering – Theory, Algorithms,* and Applications, Philadelphia, SIAM, 2007. ISBN: 978-0-898716-23-8

[HK14]: Hoffmann, Markus; Klinkenberg, Ralf: *Rapid Miner - Data Mining use cases and business analytics applications*, Boca Raton, CRC Press Verlag, 2014 ISBN: 978-1-4822-0549-7

[HKP12]: Han, Jiawei; Kamber, Micheline; Pei, Jian: *Data Mining – Concepts and Techniques*, 3. Aufl., Waltham, Morgan Kaufmann, 2012. ISBN: 978-0-12-381479-1

[KD15]: Kotu, Vijay; Deshpande, Bala: *Predictive Analytics and Data Mining – Concepts and Practice with RapidMiner*, Waltham, Morgan Kaufmann, 2015. ISBN: 978-0-12-801460-8

[Küp99]: Küpper, Bertram: *Data Mining in der Praxis – Ein Ansatz zur Nutzung der Potentiale von Data Mining im betrieblichen Umfeld*, Frankfurt am Main, Peter Lang Verlag, 1999. ISBN: 3-631-34106-7

[KZ15]: Khan, Samee U.; Zomaya, Albert Y.: Handbook on Data Centers, New York, Springer Science+Business Media Verlag, 2015. ISBN: 978-1-4939-2092-1

[LB11]: Linoff, Gordon S; Berry, Michael J.A.: *Data Mining Techniques – For Marketing, Sales, and Customer Relationship Management*, 3. Aufl., Indianapolis, Wiley, 2011. ISBN: 978-0-470-65093-6

[Leh16]: Lehmacher, Wolfgang: *Globale Supply Chain – Technischer Fortschritt, Transformation und Circular Economy*, Wiesbaden, Springer Verlag, 2016. ISBN: 978-3-658-10159-6

[MR15]: Masulli, Francesco; Rovetta, Stefano: Clustering High-Dimensional Data, in: Masulli, Francesco; Petrosino, Alfredo; Rovetta, Stefano (Hrsg.): *Clustering High-Dimensional Data – First International Workshop, CHDD 2012*, Springer Verlag, 2015. ISBN: 978-3-662-48577-4

[Net14]: Nettleton, David: *Commercial Data Mining – Processing, Analysis and Modeling for Predictive Analytics Projects*, Elsevier, Morgan Kaufmann, 2014. ISBN: 978-0-12-416602-8

[OD08]: Olson, David L; Delen, Dursun: *Advanced Data Mining Techniques*, Berlin, Heidelberg, Springer-Verlag, 2008. ISBN: 978-3-540-76916-3

[OGT04]: Orhanovic, Jens; Grodtke, Ivo; Tiefenbacher, Michael: *DB2 Administration – Einführung, Handbuch und Referenz*, München, Addison-Wesley Verlag, 2004. ISBN: 3-8273-2080-1

[Pet05]: Petersohn, Helge: *Data Mining – Verfahren, Prozesse, Anwendungsarchitektur*, München, Oldenbourg Verlag, 2005. ISBN: 978-3-486-57715-0

[Pie15]: Piegorsch, Walter: *Statistical Data Analytics – Foundations for Data Mining, Informatics, and Knowledge Discovery*, West Sussex, Wiley, 2015. ISBN: 978-1-118-61965-0

[RM15]: Rokach, Lior; Maimon, Oded: *Data Mining with decision trees – Theory and applications*, 2. Aufl., Singapore, World Scientific, 2015. ISBN: 978-9814590082

[Run15]: Runkler, Thomas A.: *Data Mining – Modelle und Algorithmen intelligenter Datenanalyse*, 2. Aufl., Wiesbaden, Springer Verlag, 2015. ISBN: 978-3-8348-2171-3

[Sha13]: Sharafi, Armin: *Knowledge Discovery in Databases – Eine Analyse des Änderungsmanagements in der Produktionsentwicklung*, Wiesbaden, Springer Gabler Verlag, 2013. ISBN: 978-3-658-02002-6

[SZT+15]: Shi, Yong; Zhang, Lingling; Tian, Yingjie; Li, Xingsen: *Intelligent Knowledge – A Study Beyond Data Mining*, Berlin, Springer Verlag, 2015. ISBN: 978-3-662-46193-8

[Tha00]: Thalheim, Bernhard: *Entity-Relationship Modeling – Foundations of Database Technology*, Berlin Heidelberg, Springer Verlag, 2000. ISBN: 3-540-65470-4

[TSK06]: Tan, Pang-Ning; Steinbach, Michael; Kumar, Vipin: *Introduction to Data Mining*, Boston, Pearson Education, 2006. ISBN: 0-321-32136-7

[WFH11]: Witten, Ian H.; Frank, Eibe; Hall, Mark A.: *Data Mining – Practical Machine Learning Tools and Techniques*, 3. Aufl., Elsevier, Morgan Kaufmann Verlag, 2011. ISBN: 978-0-12-374856-0

Zeitschrift

[FPS96]: Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic: *From Data Mining to Knowledge Discovery in Databases*, in AI Magazine 17, S. 37–54, 1996.

[Tha00]: Chen, Peter*: The Entity-Relationship Model - Toward a Unified View of Data*. In: ACM Transactions on Database Systems 1/1/1976 ACM-Press ISSN , S. 9–36. New York, USA: ACM 1976

Sammelband

[Din95]: Deutsches Institut für Normung: *DIN 44300 – Informationsverarbeitung – Begriffe*, Beuth-Verlag, Berlin, 1995

[ITP16]: Lehrstuhl für IT in Produktion und Logistik, TU Dortmund: *Whitepaper für das Thema: Vorgehensmodell zur Musterextraktion in SCs (MESC)*.

Internetquelle

[Oec16]: OECD: Glossary of statistical terms. URL: https://stats.oecd.org/glossary/detail.asp?ID=542/ - Abrufdatum: 16.08.2016

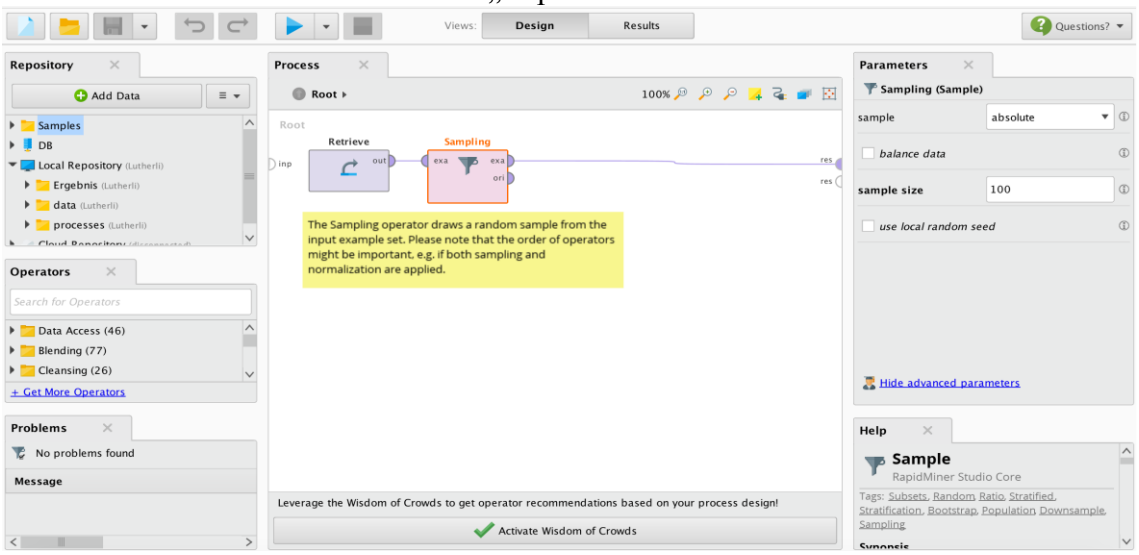
[Rap16]: Rapidminer: Rapidminer Documentation – Operator Reference Guide. URL: http://docs.rapidminer.com/studio/operators/ - Abrufdatum: 20.09.2016

[Rap12]: Rapidminer: RapidMiner 5.2 - Advanced-Charts. URL: http://rapidminer.com/wp-content/uploads/2013/10/ RapidMiner-5.2-Advanced-Charts-english-v1.0.pdf/ - Abrufdatum: 20.11.2016

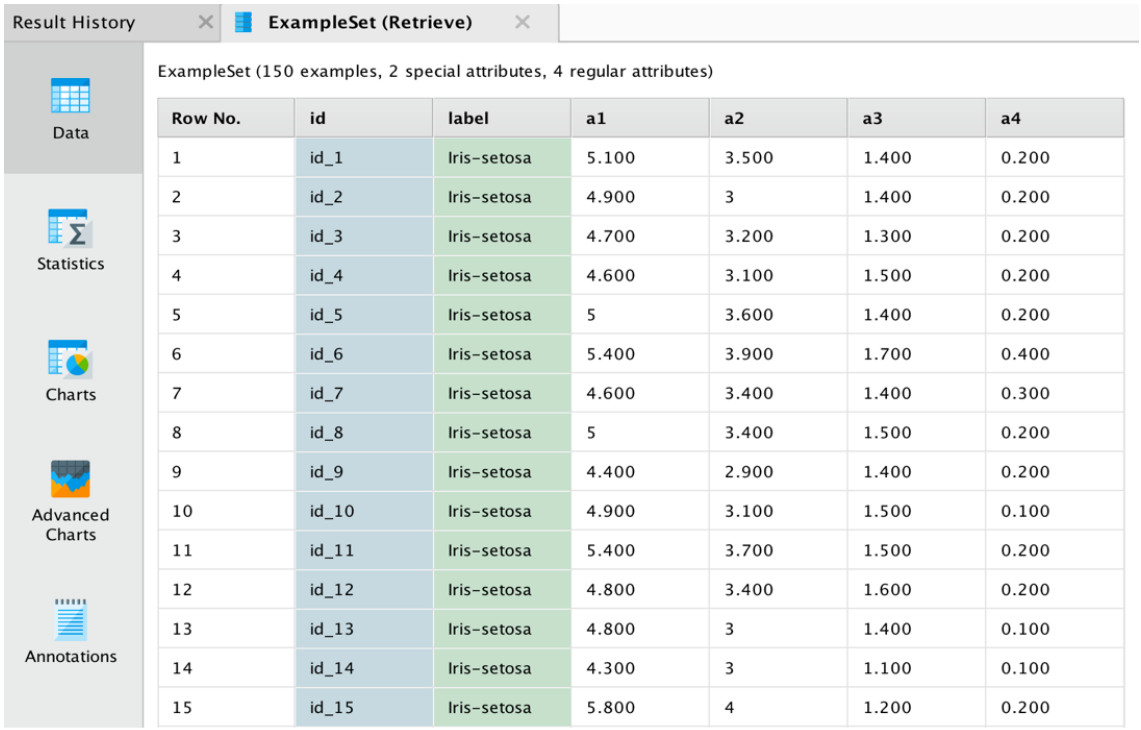
Anhang

Anhang 1: Screenshot der Software “RapidMiner”

Die Benutzeroberfläche der Software „RapidMiner

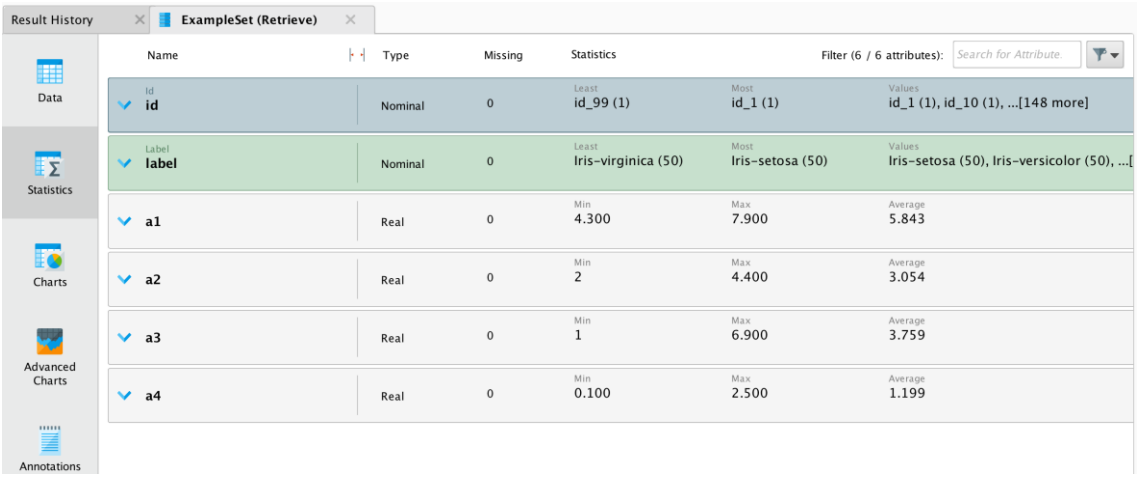


Beispielaussehen des Menüs „Datenübersicht“

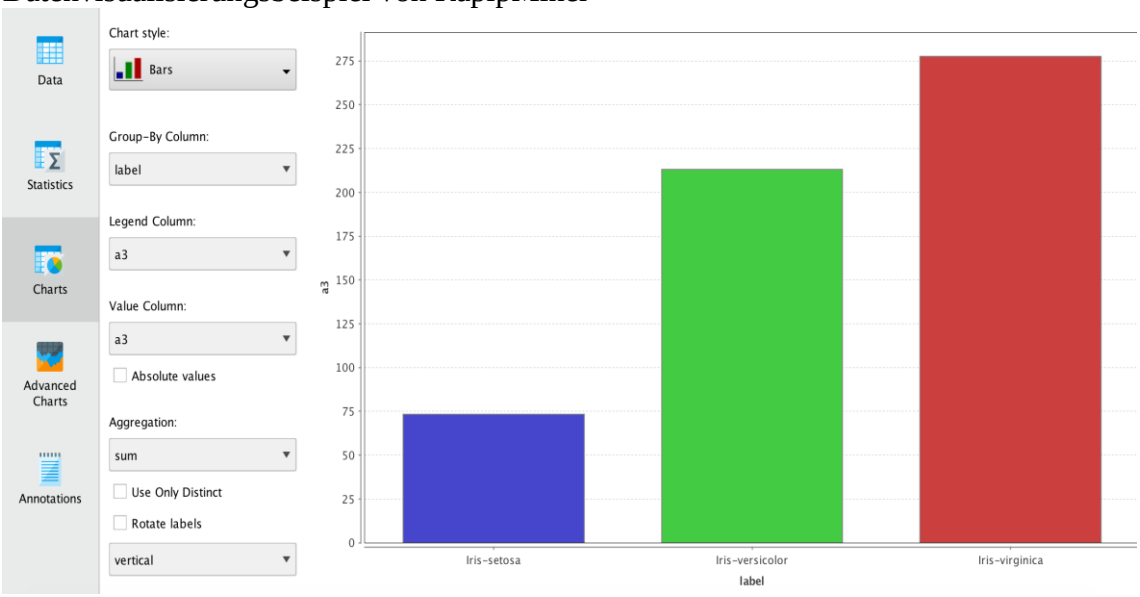


Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	3	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200
6	id_6	Iris-setosa	5.400	3.900	1.700	0.400
7	id_7	Iris-setosa	4.600	3.400	1.400	0.300
8	id_8	Iris-setosa	5	3.400	1.500	0.200
9	id_9	Iris-setosa	4.400	2.900	1.400	0.200
10	id_10	Iris-setosa	4.900	3.100	1.500	0.100
11	id_11	Iris-setosa	5.400	3.700	1.500	0.200
12	id_12	Iris-setosa	4.800	3.400	1.600	0.200
13	id_13	Iris-setosa	4.800	3	1.400	0.100
14	id_14	Iris-setosa	4.300	3	1.100	0.100
15	id_15	Iris-setosa	5.800	4	1.200	0.200

Beispielaussehen des Menüs „Datenstatistik“



Datenvisualisierungsbeispiel von RapipMiner



Anhang 2: Screenshot von den 10 Datentabellen, die im ER-Modell stehen

Screenshot von der Datentabelle“Workpiece”

Guid	CarrierId	CarrierPosition	PlantId	SystemId	LineId	Tag	ProductId	ProductVersion	OrderId	TTableId	BeginOfManufacturing	EndOfManufacturing
{728A85D1-9175-4798-B087-00003FB81B1A}	NULL	NULL	1	2	5	15492381093502500	PROD00000014	0	ORDR00000153	1	02.12.15 23:06	02.12.15 23:20
{93036358-423D-4B57-A573-0000616CB356}	NULL	NULL	1	2	5	16082391071507200	PROD00000012	0	ORDR00000144	1	22.02.16 16:51	22.02.16 17:02
{E478CCD3-7A71-4FE0-8151-0000706B8713}	NULL	NULL	1	2	1	15142355152108700	PROD00000024	0	ORDR00000024	1	31.03.15 14:36	31.03.15 14:41
{EB5C4521-3547-4F67-9582-0000AE5A96FF}	NULL	NULL	1	2	1	15432374154101900	PROD00000024	0	ORDR00000024	1	22.10.15 03:07	22.10.15 03:12
{CC05B3B9-7578-4206-912E-0000B08D17F2}	NULL	NULL	1	2	1	15482378074102700	PROD00000012	0	ORDR00000012	1	26.11.15 03:54	26.11.15 04:00
{DEF10002-6786-42B7-80FC-0000C6DBF0D6}	NULL	NULL	1	2	1	15442374073102000	PROD00000012	0	ORDR00000012	1	28.10.15 03:37	28.10.15 03:43
{0053A332-4BBC-4D5A-8AFA-0000CFFADAE1}	0	0	1	2	1	15172359152104500	PROD00000024	0	ORDR00000024	1	2015-04-21 10:26:28.080	21.04.15 10:55
{A0D68DBB-3295-45BE-BA9F-0000DE630386}	NULL	NULL	1	2	5	16132394155505600	PROD00000024	0	ORDR00000163	1	01.04.16 13:31	01.04.16 13:37
{EE843B26-DF0E-4E64-A349-0000FEC99297}	NULL	NULL	1	2	1	15122351072100500	PROD00000012	0	ORDR00000012	1	2015-03-17 01:15:01.093	17.03.15 01:25
{47D07C34-D10A-4629-A0F4-00011F590ADC}	NULL	NULL	1	2	1	15442375154110000	PROD00000024	0	ORDR00000024	1	29.10.15 22:30	29.10.15 22:35
{8D0F4429-198F-4CAC-9CE4-000155052719}	NULL	NULL	1	2	5	16042388071502800	PROD00000012	0	ORDR00000144	1	25.01.16 20:41	25.01.16 20:49
{D05F1ECF-83C0-4AB6-A119-00017385B2D5}	NULL	NULL	1	2	1	14382310155108500	PROD00000024	0	ORDR00000024	NULL	2014-09-19 19:11:17.017	19.09.14 19:39
{F480DFEE-8982-47DD-9A2B-0001A34AC5DA}	NULL	NULL	1	2	1	15272362154108400	PROD00000024	0	ORDR00000024	1	02.07.15 14:07	02.07.15 14:16
{5C05B6CC-C30D-46A8-8CD5-0001A3CBDC77}	NULL	NULL	1	2	1	15042329603104100	PROD00000019	0	ORDR00000015	1	21.01.15 07:15	21.01.15 07:22
{BC723FDD-2146-4B18-BCB1-0001A7182CDA}	NULL	NULL	1	2	1	14442310073106000	PROD00000012	0	ORDR00000012	NULL	29.10.14 11:17	29.10.14 11:23
{80E5DDD0-E969-4FD2-A9C4-0001F0F6F653}	NULL	NULL	1	2	5	15512386152508100	PROD00000024	0	ORDR00000163	1	15.12.15 15:02	15.12.15 15:11
{833C6B4C-D79A-4239-8ED6-00020F1C24E8}	NULL	NULL	1	2	1	15162358092106500	PROD00000014	0	ORDR00000010	1	2015-04-14 13:32:35.020	14.04.15 13:41
{1AB83761-0B85-4759-B635-00021FC5875D}	NULL	NULL	1	2	1	15402374594111300	PROD00000034	0	ORDR00000033	1	01.10.15 17:26	01.10.15 17:32
{FB35D0D1-2883-4CE8-8AFE-00028398D8A6}	NULL	NULL	1	2	1	15322366535109200	PROD00000066	0	ORDR00000097	1	07.08.15 20:16	2015-08-07 20:28:50.063
{8304E745-C244-4BE2-A785-0002B0D87249}	NULL	NULL	1	2	1	15252362092102700	PROD00000014	0	ORDR00000010	1	16.06.15 06:48	16.06.15 06:55
{246FC776-9086-4AA1-8CEA-0002B807DF5A}	NULL	NULL	1	2	1	15272362151102100	PROD00000024	0	ORDR00000024	1	29.06.15 03:27	29.06.15 03:37
{E3D71C31-5B64-4F22-B25F-0002D5C556A7}	NULL	NULL	1	2	1	14492326154113600	PROD00000024	0	ORDR00000024	1	04.12.14 23:09	04.12.14 23:16
{5B4BE1A7-FBCF-4FC2-9AED-0002DE5B58EA}	NULL	NULL	1	2	1	15192361155107900	PROD00000024	0	ORDR00000024	1	2015-05-08 12:28:38.093	08.05.15 12:34
{B8AB208B-AB17-4025-B0AB-0002E0BF4895}	NULL	NULL	1	2	1	14282308024102700	PROD00000007	0	ORDR00000007	NULL	11.07.14 06:06	11.07.14 06:11
{01E4B5E5-23B5-4C9F-AFFE-0002F3AFD8B9}	NULL	NULL	1	2	1	15482378152107300	PROD00000024	0	ORDR00000024	1	24.11.15 12:57	24.11.15 13:04
{8899980B-02AF-4FBC-B134-00034DB1D839}	NULL	NULL	1	2	1	15192361022100600	PROD00000007	0	ORDR00000007	1	05.05.15 02:22	05.05.15 02:34
{654603AA-9FE3-43BB-A5DE-00037A4A2A3D}	0	0	1	2	11	15452374172106900	PROD00000022	0	ORDR00000016	1	03.11.15 10:18	03.11.15 11:12
{C17F3F94-7497-4033-94CC-00039D1A03CB}	NULL	NULL	1	2	1	15312366092112100	PROD00000014	0	ORDR00000010	1	28.07.15 21:32	28.07.15 21:39
{722DB136-6C7A-4390-9921-0003ADB420FB}	NULL	NULL	1	2	1	14452311093110700	PROD00000014	0	ORDR00000010	NULL	05.11.14 20:58	05.11.14 21:11

Screenshot von der Datentabelle “OperationProtocol”

WorkpieceGuid	WorkSequence	RoutingId	RoutingVersion	RoutingSequence	PlantId	SystemId	LineId	WorkplaceId	ProcessId	BeginOfWork	EndOfWork
{728A85D1-9175-4798-B087-00003FB81B1A}	1	ROUT00000003	0	10	1	2	5	80	99	02.12.15 23:06	02.12.15 23:06
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	1	2	5	81	1	2015-12-02 23:07:06.090	02.12.15 23:07
{728A85D1-9175-4798-B087-00003FB81B1A}	3	ROUT00000003	0	30	1	2	5	82	1	02.12.15 23:09	02.12.15 23:10
{728A85D1-9175-4798-B087-00003FB81B1A}	4	ROUT00000003	0	40	1	2	5	91	1	02.12.15 23:10	02.12.15 23:10
{728A85D1-9175-4798-B087-00003FB81B1A}	5	ROUT00000003	0	45	1	2	5	92	1	02.12.15 23:10	02.12.15 23:10
{728A85D1-9175-4798-B087-00003FB81B1A}	6	ROUT00000003	0	50	1	2	5	93	1	02.12.15 23:11	02.12.15 23:11
{728A85D1-9175-4798-B087-00003FB81B1A}	7	ROUT00000003	0	60	1	2	5	102	1	2015-12-02 23:11:39.073	02.12.15 23:12
{728A85D1-9175-4798-B087-00003FB81B1A}	8	ROUT00000003	0	70	1	2	5	103	1	02.12.15 23:12	02.12.15 23:12
{728A85D1-9175-4798-B087-00003FB81B1A}	9	ROUT00000003	0	80	1	2	5	113	1	02.12.15 23:19	2015-12-02 23:19:41.077
{728A85D1-9175-4798-B087-00003FB81B1A}	10	ROUT00000003	0	85	1	2	5	120	1	02.12.15 23:19	02.12.15 23:19
{728A85D1-9175-4798-B087-00003FB81B1A}	11	ROUT00000003	0	90	1	2	5	121	1	02.12.15 23:20	02.12.15 23:20
{93036358-423D-4B57-A573-0000616CB356}	1	ROUT00000003	0	10	1	2	5	80	99	22.02.16 16:51	22.02.16 16:51
{93036358-423D-4B57-A573-0000616CB356}	2	ROUT00000003	0	20	1	2	5	81	1	22.02.16 16:51	22.02.16 16:52
{93036358-423D-4B57-A573-0000616CB356}	3	ROUT00000003	0	30	1	2	5	82	1	22.02.16 16:54	22.02.16 16:54
{93036358-423D-4B57-A573-0000616CB356}	4	ROUT00000003	0	40	1	2	5	91	1	22.02.16 16:54	22.02.16 16:55
{93036358-423D-4B57-A573-0000616CB356}	5	ROUT00000003	0	45	1	2	5	92	1	22.02.16 16:55	22.02.16 16:55
{93036358-423D-4B57-A573-0000616CB356}	6	ROUT00000003	0	50	1	2	5	93	1	22.02.16 16:55	22.02.16 16:56
{93036358-423D-4B57-A573-0000616CB356}	7	ROUT00000003	0	60	1	2	5	102	1	22.02.16 16:56	22.02.16 16:56
{93036358-423D-4B57-A573-0000616CB356}	8	ROUT00000003	0	70	1	2	5	103	1	22.02.16 16:56	22.02.16 16:57
{93036358-423D-4B57-A573-0000616CB356}	9	ROUT00000003	0	80	1	2	5	113	1	22.02.16 16:57	22.02.16 16:58
{93036358-423D-4B57-A573-0000616CB356}	10	ROUT00000003	0	85	1	2	5	120	1	22.02.16 17:02	22.02.16 17:02
{93036358-423D-4B57-A573-0000616CB356}	11	ROUT00000003	0	90	1	2	5	121	1	22.02.16 17:02	2016-02-22 17:02:48.097
{E478CCD3-7A71-4FE0-8151-0000706B8713}	1	ROUT00000003	0	10	1	2	1	80	99	31.03.15 14:36	31.03.15 14:36
{E478CCD3-7A71-4FE0-8151-0000706B8713}	2	ROUT00000003	0	20	1	2	1	81	1	31.03.15 14:34	31.03.15 14:35
{E478CCD3-7A71-4FE0-8151-0000706B8713}	3	ROUT00000003	0	30	1	2	1	82	1	2015-03-31 14:37:02.077	31.03.15 14:37
{E478CCD3-7A71-4FE0-8151-0000706B8713}	4	ROUT00000003	0	40	1	2	1	91	1	31.03.15 14:38	31.03.15 14:38
{E478CCD3-7A71-4FE0-8151-0000706B8713}	5	ROUT00000003	0	45	1	2	1	92	1	31.03.15 14:38	31.03.15 14:39
{E478CCD3-7A71-4FE0-8151-0000706B8713}	6	ROUT00000003	0	50	1	2	1	93	1	31.03.15 14:39	31.03.15 14:39
{E478CCD3-7A71-4FE0-8151-0000706B8713}	7	ROUT00000003	0	60	1	2	1	102	1	31.03.15 14:39	31.03.15 14:40

Screenshot von der Datentabelle “OperationResultProtocol”

WorkpieceGuid	WorkSequence	RoutingId	RoutingVersion	RoutingSequence	ResultSequence	ParameterDescriptionId	ParameterDe	ParameterId	ParameterVersion	ParentParameterId	ParentParameterVersion
{728A85D1-9175-4798-B087-00003FB81B1A}	1	ROUT00000003	0	10	1	PDES00000027	0	PARA00001359	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	1	ROUT00000003	0	10	2	PDES00000137	0	PARA00001361	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	1	ROUT00000003	0	10	3	PDES00000164	0	PARA00001362	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	1	ROUT00000003	0	10	4	PDES00000179	0	PARA00001363	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	1	ROUT00000003	0	10	5	PDES00000186	0	PARA00005007	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	1	PDES00000041	0	PARA00001375	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	2	PDES00000028	0	PARA00001360	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	3	PDES00000040	0	PARA00001374	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	4	PDES00000042	0	PARA00001376	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	5	PDES00000043	0	PARA00001377	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	6	PDES00000044	0	PARA00001378	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	7	PDES00000045	0	PARA00001379	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	8	PDES00000046	0	PARA00001380	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	9	PDES00000165	0	PARA00001387	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	10	PDES00000171	0	PARA00001388	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	11	PDES00000164	0	PARA00001362	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	12	PDES00000179	0	PARA00001363	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	13	PDES00000048	0	PARA00001382	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	14	PDES00000050	0	PARA00001383	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	15	PDES00000051	0	PARA00001384	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	16	PDES00000148	0	PARA00001385	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	17	PDES00000158	0	PARA00001386	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	2	ROUT00000003	0	20	18	PDES00000173	0	PARA00001389	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	3	ROUT00000003	0	30	1	PDES00000166	0	PARA00001403	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	3	ROUT00000003	0	30	2	PDES00000067	0	PARA00001396	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	3	ROUT00000003	0	30	3	PDES00000142	0	PARA00001400	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	3	ROUT00000003	0	30	4	PDES00000065	0	PARA00001394	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	3	ROUT00000003	0	30	5	PDES00000063	0	PARA00001392	0	NULL	NULL
{728A85D1-9175-4798-B087-00003FB81B1A}	3	ROUT00000003	0	30	6	PDES00000064	0	PARA00001393	0	NULL	NULL

Screenshot von der Datentabelle “ParameterDescription”

ParameterDescriptionID	ParameterDescriptionVersion	ParentParameterDescriptionId	ParentParameterDescriptionVersion	IndentLevel	ClassAssociatedTo	Category	PlantId	SystemId	LineId	Name	Description1
PDES00000024	0	NULL	0	1	ProductEntity	NTM	NULL	NULL	NULL	NTM_Processdata	Product related processdata
PDES00000027	0	PDES00000024	0	2	ProductEntity	NTM	NULL	NULL	NULL	SAP-No.	SAP order number
PDES00000028	0	PDES00000024	0	2	ProductEntity	NTM	NULL	NULL	NULL	Serial no. BE	Serial number back end
PDES00000031	0	PDES00000038	0	1	ProductEntity	NTM	NULL	NULL	NULL	Power voltage	Nominal supply voltage
PDES00000032	0	PDES00000038	0	1	ProductEntity	NTM	NULL	NULL	NULL	Power frequency	Nominal supply frequency
PDES00000033	0	PDES00000038	0	1	ProductEntity	NTM	NULL	NULL	NULL	Connector-ID	Connector (power cord) ID
PDES00000034	0	PDES00000038	0	1	ProductEntity	NTM	NULL	NULL	NULL	min. Voltage	Minimal supply voltage
PDES00000035	0	PDES00000038	0	1	ProductEntity	NTM	NULL	NULL	NULL	max. Voltage	Maximal supply voltage
PDES00000036	0	PDES00000038	0	1	ProductEntity	NTM	NULL	NULL	NULL	min. Frequency	Minimal supply frequency
PDES00000037	0	PDES00000038	0	1	ProductEntity	NTM	NULL	NULL	NULL	max. Frequency	Maximal supply frequency
PDES00000038	0	NULL	NULL	NULL	ProductEntity	NTM	NULL	NULL	NULL	NTM_Parameter	Product related parameter
PDES00000039	0	NULL	NULL	NULL	ProductEntity	WP17.1	NULL	NULL	NULL	WP17_1_Processdata	WP17.1 related processdata
PDES00000040	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Bowl-Temperature	Bowl temperature (measure)
PDES00000041	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Errorstate	Errorstate
PDES00000042	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Motorspeed	Motor speed (measure)
PDES00000043	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Motorcurrent	Motor current (measure)
PDES00000044	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Mototrtemperature	Motor temperature (measure)
PDES00000045	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	DI-Test 1 PASS	DI-test 1 (low current)
PDES00000046	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	DI-Test 2 PASS	DI-test 2 (high current)
PDES00000047	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Safety Test PASS	Safety Test
PDES00000048	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Failed Teststeps	List of failed test steps
PDES00000050	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Power Voltage	Supply voltage (measure)
PDES00000051	0	PDES00000039	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	Power Frequency	Supply frequency (measure)
PDES00000052	0	NULL	NULL	NULL	ProductEntity	WP17.1	NULL	NULL	NULL	WP17_1_Parameter	WP17.1 related parameter
PDES00000053	0	PDES00000052	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	min. Bowl Temp.	Minimal bowl temperature
PDES00000054	0	PDES00000052	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	max. Bowl Temp.	Maximum bowl temperature
PDES00000055	0	PDES00000052	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	min. Motorspeed	Minimum motor speed
PDES00000056	0	PDES00000052	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	max. Motorspeed	Maximum motor speed
PDES00000057	0	PDES00000052	0	1	ProductEntity	WP17.1	NULL	NULL	NULL	min. Motorcurrent	Minimum motor current

Screenshot von der Datentabelle “Order”

OrderId	OrderNumber	ProductId	ProductVersion	Planned	PlannedScrapRate	TotalIn	TotalOut	Pass	Fail	Scrap	Delivered	DueDate	Status	PlantId	SystemId	LineId	TimeStamp
ORDR00000006	Order 0001	PROD00000006	0	100000	0	17388	5495	4799	696	1	4799	2013-03-13 00:00:00.000	ONHOLD	1	2	1	27.04.15 09:59
ORDR00000007	Order 0002	PROD00000007	0	100000	0	12576	12224	11989	235	1	11989	NULL	ONHOLD	1	2	1	01.02.16 11:34
ORDR00000008	test	PROD00000008	0	100000	0	2570	1300	1284	16	NULL	1284	NULL	ONHOLD	1	2	1	18.02.16 16:51
ORDR00000009	Order 0004.1	PROD00000009	0	1	0	912	886	875	11	1	875	2014-06-11 00:00:00.000	ONHOLD	1	2	1	11.03.16 06:18
ORDR00000010	Order 0009	PROD00000014	0	100000	0	62786	62221	61471	750	1	61471	NULL	ONHOLD	1	2	1	29.03.16 08:34
ORDR00000011	Mexico	PROD00000013	0	999999	0	11934	11771	11621	150	NULL	11621	NULL	ONHOLD	1	2	1	29.01.16 16:25
ORDR00000012	Poland	PROD00000012	0	999999	0	62956	62473	61895	578	NULL	61895	NULL	ONHOLD	1	2	1	16.03.16 17:39
ORDR00000013	Cyprus	PROD00000018	0	999999	0	3782	3737	3667	70	NULL	3667	NULL	ONHOLD	1	2	1	24.02.16 10:07
ORDR00000014	UK	PROD00000011	0	999999	0	14412	14282	14133	149	NULL	14133	NULL	ONHOLD	1	2	1	07.04.16 08:44
ORDR00000015	Russia	PROD00000019	0	999999	0	5330	5267	5143	124	NULL	5143	NULL	ONHOLD	1	2	1	18.03.16 11:25
ORDR00000016	Austria	PROD00000022	0	999999	0	12718	12613	12497	116	NULL	12497	NULL	ONHOLD	1	2	1	05.04.16 23:02
ORDR00000017	Benelux DE	PROD00000020	0	999999	0	1008	916	910	6	NULL	910	NULL	ONHOLD	1	2	1	13.11.15 18:57
ORDR00000018	Benelux FR	PROD00000021	0	999999	0	3822	3780	3731	49	NULL	3731	NULL	ONHOLD	1	2	1	14.11.15 00:52
ORDR00000019	Benelux NL	PROD00000023	0	1	0	8777	8729	8636	93	NULL	8636	NULL	ONHOLD	1	2	1	10.12.15 07:34
ORDR00000020	Czech	PROD00000010	0	999999	0	5902	5868	5795	73	NULL	5795	NULL	ONHOLD	1	2	1	06.04.16 10:29
ORDR00000021	Switzerland DE	PROD00000015	0	999999	0	8444	8393	8322	71	NULL	8322	NULL	ONHOLD	1	2	1	2016-03-21 19:50:05.000
ORDR00000022	Switzerland FR	PROD00000016	0	999999	0	5367	5293	5259	34	NULL	5259	NULL	ONHOLD	1	2	1	21.03.16 12:35
ORDR00000023	Switzerland IT	PROD00000017	0	999999	0	3949	3877	3826	51	NULL	3826	NULL	ONHOLD	1	2	1	24.09.15 23:17
ORDR00000024	Spain	PROD00000024	0	999999	0	186367	185103	183219	1884	3	183219	NULL	ONHOLD	1	2	1	16.03.16 11:36
ORDR00000025	Kasachstan	PROD00000025	0	999999	0	869	858	832	26	NULL	832	NULL	ONHOLD	1	2	1	22.10.14 20:45
ORDR00000026	Baltikum	PROD00000026	0	999999	0	941	938	921	17	NULL	921	NULL	ONHOLD	1	2	1	23.07.15 15:23
ORDR00000027	Israel	PROD00000027	0	999999	0	276	29	25	4	NULL	25	NULL	ONHOLD	1	2	1	23.03.15 10:27
ORDR00000028	Italy	PROD00000028	0	999999	0	5940	5888	5839	49	NULL	5839	NULL	ONHOLD	1	2	1	05.04.16 08:11
ORDR00000029	Brazil 230V	PROD00000030	0	999999	0	252	247	244	3	NULL	244	NULL	ONHOLD	1	2	1	15.01.16 18:13
ORDR00000030	Chile	PROD00000031	0	999999	0	2715	2703	2658	45	NULL	2658	NULL	ONHOLD	1	2	1	19.03.16 06:52
ORDR00000031	Hungary	PROD00000032	0	999999	0	359	353	353	0	NULL	353	NULL	ONHOLD	1	2	1	08.10.15 21:21
ORDR00000032	Morocco	PROD00000033	0	999999	0	4447	4422	4394	28	NULL	4394	NULL	ONHOLD	1	2	1	16.11.15 08:44
ORDR00000033	Romania	PROD00000034	0	999999	0	954	949	941	8	NULL	941	NULL	ONHOLD	1	2	1	26.11.15 14:24
ORDR00000034	South Africa	PROD00000035	0	999999	0	1823	1797	1781	16	NULL	1781	NULL	ONHOLD	1	2	1	10.03.16 21:09

Screenshot von der Datentabelle “Line ID”

LineId	SystemId	PlantId	Name	Class	Type	Description1	Description2	EquipmentCo	CreatedNam	CreatedDate
1	2	1	Line1	Assembly	NULL	Linie TM 1			AES	02.11.15 08:25
5	2	1	Linie5	Assembly		Linie TM 5			AES	02.11.15 08:25
11	2	1	Handarbeitsp	Assembly		Handarbeitsplatz			Krane	12.11.14 14:12

Screenshot von der Datentabelle “Product”

ProductId	Version	Class	PlantId	SystemId	PartNumber1	PartNumber2	PartNumber3	PartNumber4	Name	Description1	Description2	ShortIdNumber	RoutingId	RoutingVersion	LabelId
PROD00000004	0	NTM	NULL	NULL	Product1	NULL	NULL	PROD01	Product1	<Beschreibung eingeben>		1	ROUT00000003	0	LABL00000001
PROD00000006	0	NTM	0	0	1	NULL	NULL	1	Germany	NTM TypeNo. 01		2	ROUT00000003	0	LABL00000001
PROD00000007	0	NTM	0	0	2	NULL	NULL	2	Taiwan	120V		3	ROUT00000003	0	LABL00000001
PROD00000008	0	NTM	0	0	3	NULL	NULL	3	UAE + koweit + qatar + bahrain	3		4	ROUT00000003	0	LABL00000001
PROD00000009	0	NTM	0	0	4	NULL	NULL	4	Angola	Angola		5	ROUT00000003	0	LABL00000001
PROD00000010	0	NTM	0	0	5	NULL	NULL	5	Czech	Czech		6	ROUT00000003	0	LABL00000001
PROD00000011	0	NTM	0	0	6	NULL	NULL	6	UK	UK		7	ROUT00000003	0	LABL00000001
PROD00000012	0	NTM	0	0	7	NULL	NULL	7	Poland	Poland		8	ROUT00000003	0	LABL00000001
PROD00000013	0	NTM	0	0	8	NULL	NULL	8	Mexico	Mexico		9	ROUT00000003	0	LABL00000001
PROD00000014	0	NTM	0	0	9	NULL	NULL	9	Portugal	Portugal		10	ROUT00000003	0	LABL00000001
PROD00000015	0	NTM	0	0	10	NULL	NULL	10	Switzerland DE	Switzerland DE		11	ROUT00000003	0	LABL00000001
PROD00000016	0	NTM	0	0	11	NULL	NULL	11	Switzerland FR	Switzerland FR		12	ROUT00000003	0	LABL00000001
PROD00000017	0	NTM	0	0	12	NULL	NULL	12	Switzerland IT	Switzerland IT		13	ROUT00000003	0	LABL00000001
PROD00000018	0	NTM	0	0	33	NULL	NULL	33	Cyprus	Cyprus		14	ROUT00000003	0	LABL00000001
PROD00000019	0	NTM	0	0	60	NULL	NULL	60	Russia	Russia		15	ROUT00000003	0	LABL00000001
PROD00000020	0	NTM	0	0	20	NULL	NULL	20	Benelux DE	Benelux DE		16	ROUT00000003	0	LABL00000001
PROD00000021	0	NTM	0	0	21	NULL	NULL	21	Benelux FR	Benelux FR		17	ROUT00000003	0	LABL00000001
PROD00000022	0	NTM	0	0	17	NULL	NULL	17	Austria	Austria		18	ROUT00000003	0	LABL00000001
PROD00000023	0	NTM	0	0	22	NULL	NULL	22	Benelux NL	Benelux NL		19	ROUT00000003	0	LABL00000001
PROD00000024	0	NTM	0	0	15	NULL	NULL	15	Spain	Spain		20	ROUT00000003	0	LABL00000001
PROD00000025	0	NTM	0	0	46	NULL	NULL	46	Kazakhstan	Kazakhstan		21	ROUT00000003	0	LABL00000001
PROD00000026	0	NTM	0	0	19	NULL	NULL	19	Baltikum	Baltikum		22	ROUT00000003	0	LABL00000001
PROD00000027	0	NTM	0	0	44	NULL	NULL	44	Israel	Israel		23	ROUT00000003	0	LABL00000001
PROD00000028	0	NTM	0	0	14	NULL	NULL	14	Italy	Italy		24	ROUT00000003	0	LABL00000001
PROD00000029	0	NTM	0	0	23	NULL	NULL	23	Brazil 120V	Brazil 120V		25	ROUT00000003	0	LABL00000001
PROD00000030	0	NTM	0	0	24	NULL	NULL	24	Brazil 230V	Brazil 230V		26	ROUT00000003	0	LABL00000001
PROD00000031	0	NTM	0	0	30	NULL	NULL	30	Chile	Chile		27	ROUT00000003	0	LABL00000001
PROD00000032	0	NTM	0	0	41	NULL	NULL	41	Hungary	Hungary		28	ROUT00000003	0	LABL00000001
PROD00000033	0	NTM	0	0	55	NULL	NULL	55	Morocco 230V	Morocco 230V		29	ROUT00000003	0	LABL00000001

Screenshot von der Datentabelle “WorkPlace”

WorkplaceId	LineId	SystemId	PlantId	Name	Class	Type	Description1	Description2	EquipmentCode	CreatedName	CreatedDate
80	1	2	1	WP17.0	Automatic	NULL	WP17.0 Switch ON NTM			AES	29.10.13 16:37
80	5	2	1	WP17.0	Automatic	NULL	WP17.0 Switch ON NTM			AES	2015-02-11 00:00:00.000
80	11	2	1	WP17.0	Automatic	NULL	WP17.0 Switch ON NTM			AES	29.10.13 16:37
81	1	2	1	WP17.1	Automatic	NULL	WP17.1 safety check			Korn	15.01.13 14:10
81	5	2	1	WP17.1	Automatic	NULL	WP17.1 safety check			AES	2015-02-11 00:00:00.000
81	11	2	1	WP17.1	Automatic	NULL	WP17.1 safety check			Korn	29.10.13 16:37
82	1	2	1	WP17.2	Manual	NULL	WP17.2 insert bowl + boot device			Korn	2013-01-15 14:10:59.083
82	5	2	1	WP17.2	Manual	NULL	WP17.2 insert bowl + boot device			AES	2015-02-11 00:00:00.000
82	11	2	1	WP17.2	Manual	NULL	WP17.2 insert bowl + boot device			Korn	29.10.13 16:37
91	1	2	1	WP18.0	Automatic	NULL	WP18.0 HV Test + EC Test	Hight voltage test AND earth continuity test		Korn	15.01.13 14:18
91	5	2	1	WP18.0	Automatic	NULL	WP18.0 HV Test + EC Test	Hight voltage test AND earth continuity test		AES	2015-02-11 00:00:00.000
91	11	2	1	WP18.0	Automatic	NULL	WP18.0 HV Test + EC Test	Hight voltage test AND earth continuity test		Korn	29.10.13 16:37
92	1	2	1	WP18.1	Automatic	NULL	WP18.1 Switch ON NTM			AES	29.10.13 16:37
92	5	2	1	WP18.1	Automatic	NULL	WP18.1 Switch ON NTM			AES	2015-02-11 00:00:00.000
92	11	2	1	WP18.1	Automatic	NULL	WP18.1 Switch ON NTM			AES	29.10.13 16:37
93	1	2	1	WP18.2	Automatic	NULL	WP18.2 calibration			Korn	15.01.13 14:18
93	5	2	1	WP18.2	Automatic	NULL	WP18.2 calibration			AES	2015-02-11 00:00:00.000
93	11	2	1	WP18.2	Automatic	NULL	WP18.2 calibration			Korn	29.10.13 16:37
102	1	2	1	WP19.2	Automatic	NULL	WP19.2 flashing 1			Korn	15.01.13 14:19
102	5	2	1	WP19.2	Automatic	NULL	WP19.2 flashing 1			AES	2015-02-11 00:00:00.000
102	11	2	1	WP19.2	Automatic	NULL	WP19.2 flashing 1			Korn	29.10.13 16:37
103	1	2	1	WP19.3	Automatic	NULL	WP19.3 flashing 2			Korn	15.01.13 14:19
103	5	2	1	WP19.3	Automatic	NULL	WP19.3 flashing 2			AES	2015-02-11 00:00:00.000
103	11	2	1	WP19.3	Automatic	NULL	WP19.3 flashing 2			Korn	29.10.13 16:37
113	1	2	1	WP20.3	Automatic	NULL	WP20.3 laser marking			Korn	15.01.13 14:20
113	5	2	1	WP20.3	Automatic	NULL	WP20.3 laser marking			AES	2015-02-11 00:00:00.000
113	11	2	1	WP20.3	Automatic	NULL	WP20.3 laser marking			Korn	29.10.13 16:37
120	1	2	1	WP21.1	Automatic	NULL	WP21.1 DMC-Check			AES	29.10.13 16:36
120	5	2	1	WP21.1	Automatic	NULL	WP21.1 DMC-Check			AES	2015-02-11 00:00:00.000

Screenshot von der Datentabelle “Process”

ProcessId	WorkplaceId	LineId	SystemId	PlantId	Name	Class	Type	Description1	Description2	EquipmentCo	UseCarrierIdentity	UseWorkpieceIdentity	CreatedName
1	91	5	2	1	HV+EC Test	Automatic	PROCESS	Hight voltage and earth continuity test			False	True	AES
1	91	11	2	1	HV+EC Test	Automatic	PROCESS	Hight voltage and earth continuity test			False	True	Korn
1	92	1	2	1	Switch ON NTM	Automatic	PROCESS	Switch ON the NTM device			False	True	AES
1	92	5	2	1	Switch ON NTM	Automatic	PROCESS	Switch ON the NTM device			False	True	AES
1	92	11	2	1	Switch ON NTM	Automatic	PROCESS	Switch ON the NTM device			False	True	AES
1	93	1	2	1	Calibration	Automatic	PROCESS	Calibration			False	True	Korn
1	93	5	2	1	Calibration	Automatic	PROCESS	Calibration			False	True	AES
1	93	11	2	1	Calibration	Automatic	PROCESS	Calibration			False	True	Korn
1	102	1	2	1	Flashing 1	Automatic	PROCESS	Write data to flash, Part 1			False	True	Korn
1	102	5	2	1	Flashing 1	Automatic	PROCESS	Write data to flash, Part 1			False	True	AES
1	102	11	2	1	Flashing 1	Automatic	PROCESS	Write data to flash, Part 1			False	True	Korn
1	103	1	2	1	Flashing 2	Automatic	PROCESS	Write data to flash, Part 2			False	True	Korn
1	103	5	2	1	Flashing 2	Automatic	PROCESS	Write data to flash, Part 2			False	True	AES
1	103	11	2	1	Flashing 2	Automatic	PROCESS	Write data to flash, Part 2			False	True	Korn
1	113	1	2	1	Marking	Automatic	PROCESS	Laser marking			False	True	Korn
1	113	5	2	1	Marking	Automatic	PROCESS	Laser marking			False	True	AES
1	113	11	2	1	Marking	Automatic	PROCESS	Laser marking			False	True	Korn
1	120	1	2	1	DMC-Check	Automatic	PROCESS	Check the printed DMC			False	True	AES
1	120	5	2	1	DMC-Check	Automatic	PROCESS	Check the printed DMC			False	True	AES
1	120	11	2	1	DMC-Check	Automatic	PROCESS	Check the printed DMC			False	True	AES
1	121	1	2	1	Robot	Automatic	PROCESS	OK-Part handling			False	True	Korn
1	121	5	2	1	Robot	Automatic	PROCESS	OK-Part handling			False	True	AES
1	121	11	2	1	Robot	Automatic	PROCESS	OK-Part handling			False	True	Korn
1	143	1	2	1	Safety check	Manual	PROCESS	Safety check			False	True	Korn
1	143	5	2	1	Safety check	Manual	PROCESS	Safety check			False	True	AES
1	143	11	2	1	Safety check	Manual	PROCESS	Safety check			False	True	Korn
99	80	1	2	1	AES-Start	Automatic	PROCESS	Create workpiece at the pre stopper of WP17			False	True	AES
99	80	5	2	1	AES-Start	Automatic	PROCESS	Create workpiece at the pre stopper of WP17			False	True	AES
99	80	11	2	1	AES-Start	Automatic	PROCESS	Create workpiece at the pre stopper of WP17			False	True	AES

Screenshot von der Datentabelle “Traceability Data”

Identifier	Site	Line	MaterialNo	SerialNo	BatchNo	ExtHandlingUnitIdent	KindExtHandlingUnit	StockQualifikation	EAN11	StockReceiptDate	StockReceiptNumber	StockReceiptPosition	ReferenceNr	SupplierAccountNr	SupplierBatchNr	RecordInsertDate
4886328			47641	1502VE41193301								0				13.01.15 15:45
4886329			50451	FEAN:1450PA4210								0				13.01.15 15:47
4886330			47460	BE:1450PA330368								0				13.01.15 15:47
4886331			47319	1502GD61360203								0				13.01.15 15:49
4886332			47272	1502CX32114302								0				13.01.15 15:45
4886333			49868	FEAN:1449PA7223								0				13.01.15 15:47
4886334			47460	BE:1441PA133718								0				13.01.15 15:47
4886335	1000	1	61289	1,50323E+16								0				13.01.15 15:48
4886335	1000	1	61289	1,50323E+16								0				13.01.15 15:48
4886336			47645									0			1440LE6	13.01.15 15:48
4886337			48012	<Aucun code Dat								0				13.01.15 15:50
4886338			47302									0			DEFAULT	13.01.15 15:50
4886339																13.01.15 15:47
4886340			47460	BE:1440PA532858								0				13.01.15 15:47
4886341			47290	1449SB52494302								0				13.01.15 15:47
4886342			48012	1443AL41367421								0				13.01.15 15:47
4886343			49320									0			01. Mär	13.01.15 15:47
4886344			47641	1447VE31165601								0				13.01.15 15:47
4886345			47265									0			DEFAULT	13.01.15 15:49
4886346			48012	1436AL41550321								0				13.01.15 15:45
4886347			47302									0			DEFAULT	13.01.15 15:45
4886348			47272	1502CX23050902								0				13.01.15 15:50
4886349			47641	1502VE41159601								0				13.01.15 15:50
4886350			47265									0			DEFAULT	13.01.15 15:45
4886351	2001	2003	61268	1,50323E+16								0				13.01.15 15:45
4886351	2001	2003	61268	1,50323E+16								0				13.01.15 15:45
4886352			47645									0			1441LE2	13.01.15 15:45
4886353			47290	DEFAULT								0				13.01.15 15:50
4886354	2001	2001	61166	1,50323E+16								0				13.01.15 15:50

Anhang 3: Vorstellung der Software „SQL“ und der Datenexportprozess

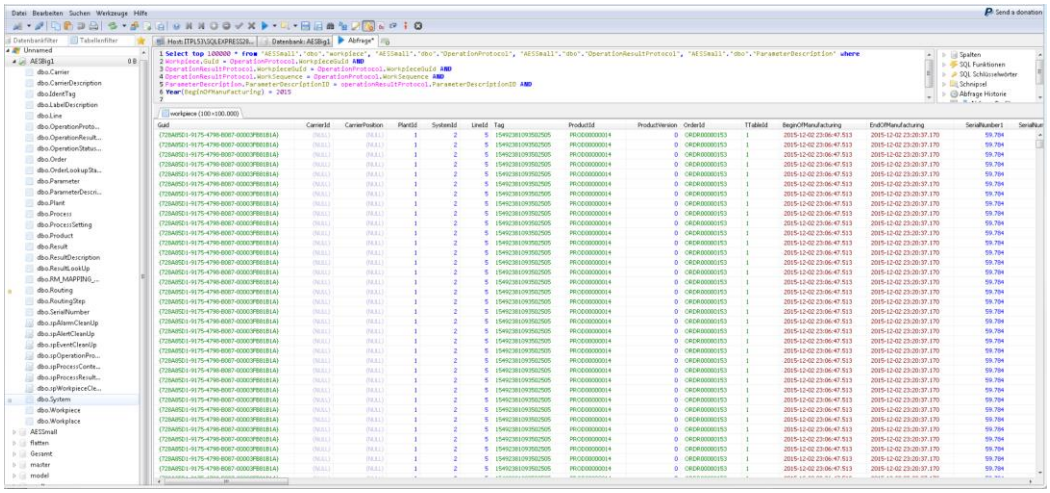


Abbildung : Aussehen der Software „SQL“

In der linken Seite des Screenshots steht eine Verzeichnisstruktur von den unterschiedlichen Datenbeständen. Jeder Datenbestand hat viele Datentabellen und vor den Namen steht eine Vorsilbe „dbo.“. Ganz oben stehen drei Dialogfelde zur Verfügung. Mithilfe des mittleren Dialogfeldes können die Datentabelle der Datenbank durch die Auswahl der Datentabelle im linken Verzeichnisstruktur angeschaut werden. Beim rechten Dialogfeld können manuell geschriebene Befehle eingegeben werden, damit die benötigten Daten nach eigenen festgelegten Bedingungen ausgewählt und exportiert werden können.
Nach dem Bedarf des späteren DM-Prozesses sollen vier Haupttabellen mithilfe der gemeinsamen Attributen „WorkpieceGuid“ und „ParameterDescriptionId“ zusammen verknüpft werden.
Die benötige Befehle wird im Anhang 4 angezeigt.
Nach der Durchführung des Befehls werden eine Datentabelle mit 100.000 Zeile Daten erstellt.
Der Export der durch die Abfrage generierten Daten kann mithilfe einer Option „Export“ realisiert werden, die im ganz oberes Menü „Werkzeug“ steht.

Anhang 4: Befehle von der Software „SQL“ zum Export von den benötigen Datentabellen zur Datenanalyse

Befehle zum Export von den „Top 100.000“ Datenzeilen der HDT

Select *top 100000 * **from** "AESBig1"."dbo"."workpiece", "AESBig1"."dbo"."OperationProtocol", "AESBig1"."dbo"."OperationResultProtocol", "AESBig1"."dbo"."ParameterDescription"* ***where***
Workpiece.Guid = OperationProtocol.WorkpieceGuid ***AND***
OperationResultProtocol.WorkpieceGuid = OperationProtocol.WorkpieceGuid ***AND***
OperationResultProtocol.WorkSequence = OperationProtocol.WorkSequence ***AND***
ParameterDescription.ParameterDescriptionID = operationResultProtocol.ParameterDescriptionID ***AND***
Year(BeginOfManufacturing) = 2015

Befehle zum Export von den „Last 100.000“ Datenzeilen der HDT

*Select top 100000 * from "AESBig1"."dbo"."workpiece", "AESBig1"."dbo"."OperationProtocol", "AESBig1"."dbo"."OperationResultProtocol", "AESBig1"."dbo"."ParameterDescription" where*
Workpiece.Guid = OperationProtocol.WorkpieceGuid AND
OperationResultProtocol.WorkpieceGuid = OperationProtocol.WorkpieceGuid AND
OperationResultProtocol.WorkSequence = OperationProtocol.WorkSequence AND
ParameterDescription.ParameterDescriptionID = operationResultProtocol.ParameterDescriptionID AND
Year(BeginOfManufacturing) = 2015
order by Guid DESC

Befehle zum Export der Datentabelle zur Datenanalyse über den Attributwert „TotalResult=FAIL“

*Select top 100000 * from "AESBig1"."dbo"."workpiece", "AESBig1"."dbo"."OperationProtocol", "AESBig1"."dbo"."OperationResultProtocol", "AESBig1"."dbo"."ParameterDescription" where*
Workpiece.Guid = OperationProtocol.WorkpieceGuid AND
OperationResultProtocol.WorkpieceGuid = OperationProtocol.WorkpieceGuid AND
OperationResultProtocol.WorkSequence = OperationProtocol.WorkSequence AND
ParameterDescription.ParameterDescriptionID = operationResultProtocol.ParameterDescriptionID AND
Year(BeginOfManufacturing) = 2015 AND
workpiece.TotalResult = 'FAIL'

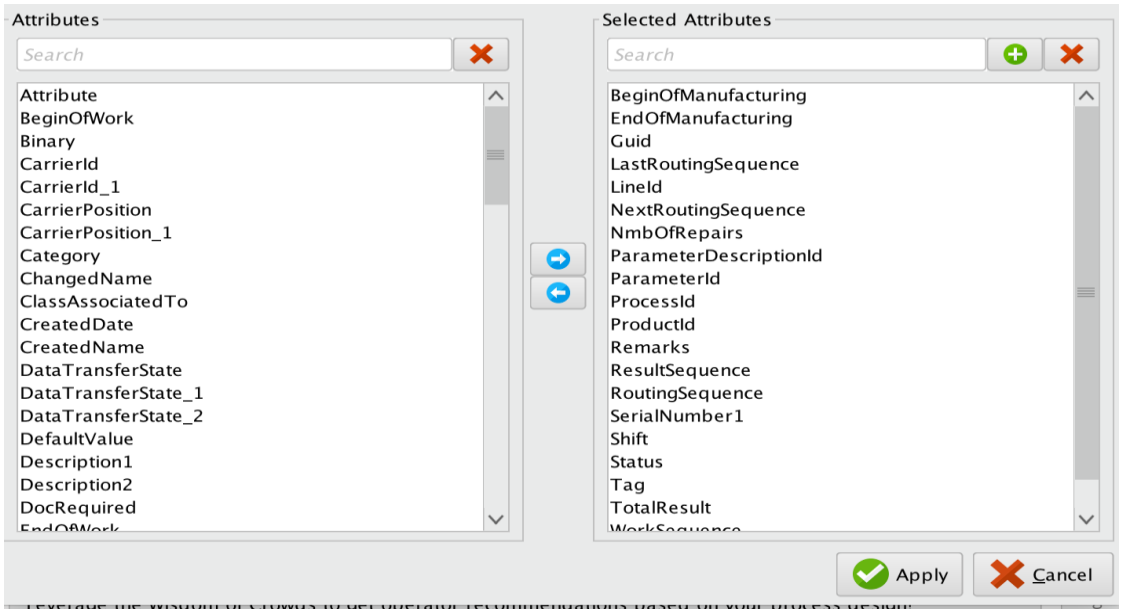
Befehle zum Export der Datentabelle zur Datenanalyse über den Attributwert „NmbOfRepairs=high repair“

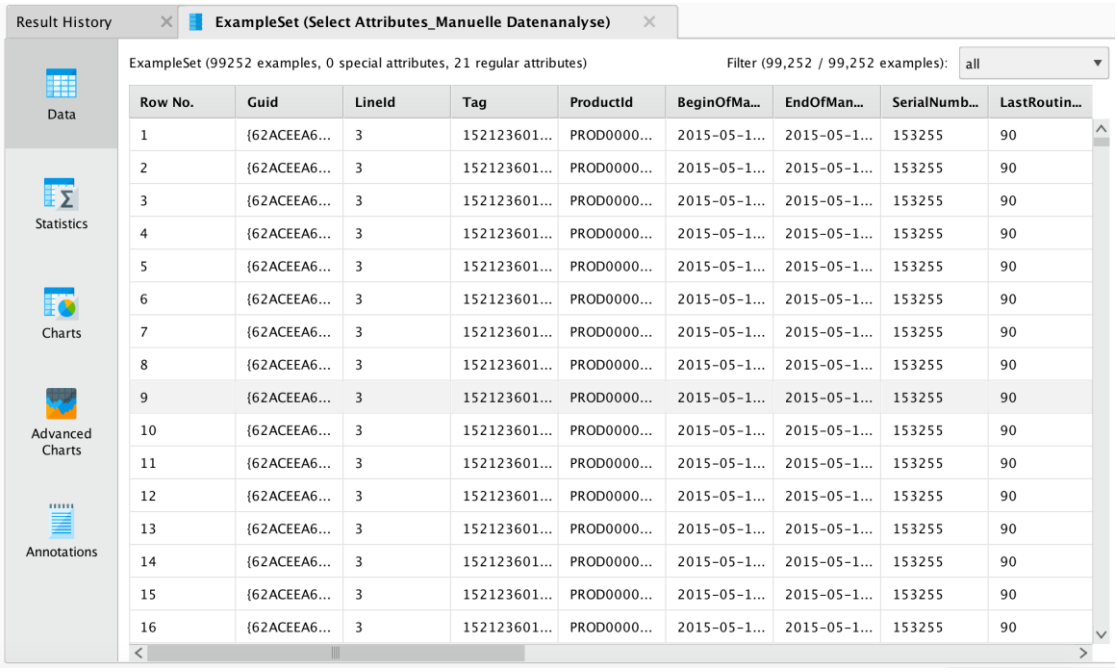
*Select top 100000 * from "AESBig1"."dbo"."workpiece", "AESBig1"."dbo"."OperationProtocol", "AESBig1"."dbo"."OperationResultProtocol", "AESBig1"."dbo"."ParameterDescription" where*
Workpiece.Guid = OperationProtocol.WorkpieceGuid AND
OperationResultProtocol.WorkpieceGuid = OperationProtocol.WorkpieceGuid AND
OperationResultProtocol.WorkSequence = OperationProtocol.WorkSequence AND
ParameterDescription.ParameterDescriptionID = operationResultProtocol.ParameterDescriptionID AND
Year(BeginOfManufacturing) = 2015 AND
Workpiece.NmbOfRepairs between '5' and '7'

Befehle zum Export der Datentabelle zur Datenanalyse über den Attributwert „NmbOfRepairs=NmbOfRepairs more than 7“

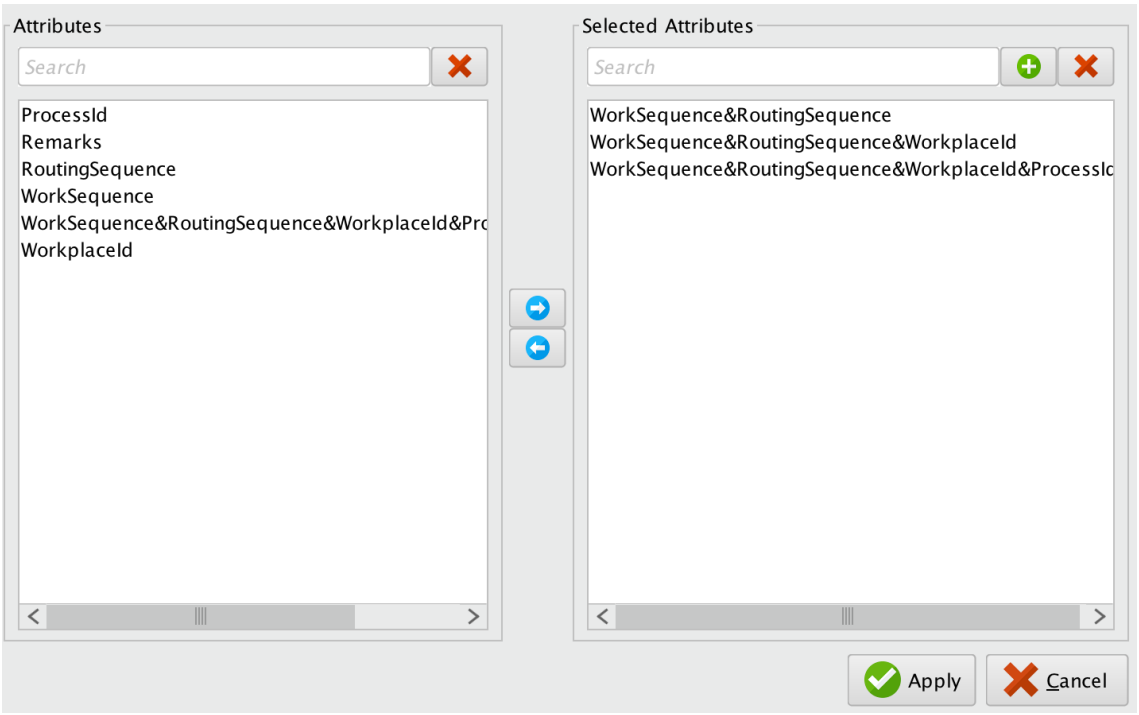
*Select top 100000 * from "AESBig1"."dbo"."workpiece", "AESBig1"."dbo"."OperationProtocol", "AESBig1"."dbo"."OperationResultProtocol", "AESBig1"."dbo"."ParameterDescription" where*
Workpiece.Guid = OperationProtocol.WorkpieceGuid AND
OperationResultProtocol.WorkpieceGuid = OperationProtocol.WorkpieceGuid AND
OperationResultProtocol.WorkSequence = OperationProtocol.WorkSequence AND
ParameterDescription.ParameterDescriptionID = operationResultProtocol.ParameterDescriptionID AND
Year(BeginOfManufacturing) = 2015 AND
Workpiece.NmbOfRepairs >= '8'

Anhang 5: Parametereinstellung und Ergebnisanzeige vom Experimentprozess „Bereinigung der Redundanzattribute “



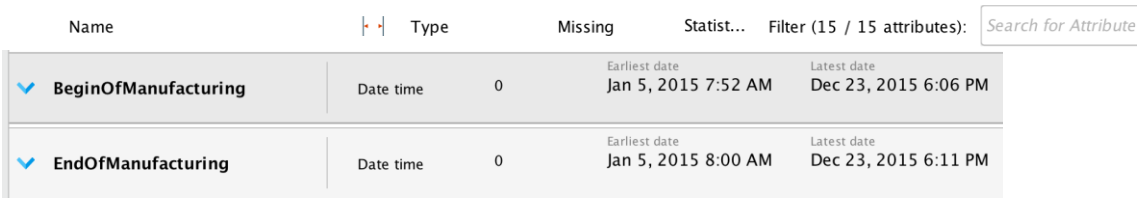


Anhang 6: Screenshot des Experimentprozesses der Aggregation der Attribute

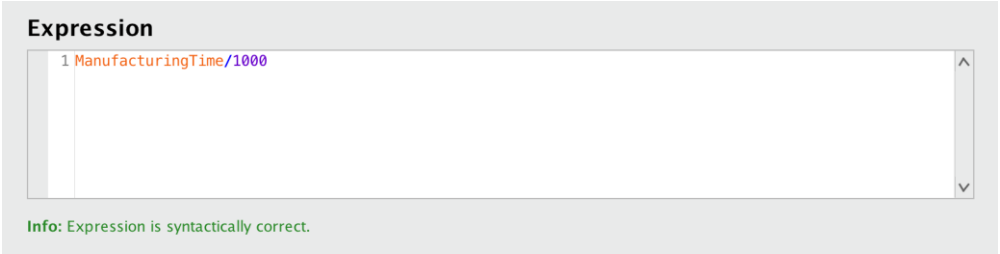


WorkSeque...	RoutingSeq...	WorkplaceId	ProcessId	Remarks	WorkSequence&RoutingSequence&WorkplaceId&ProcessId&Remarks
1	10	80	99	AESStart:...	1.0&10.0&80.0&99.0&AESStart: PASS
2	20	81	1	Safety ch...	2.0&20.0&81.0&1.0&Safety check: PASS
2	20	81	1	Safety ch...	2.0&20.0&81.0&1.0&Safety check: PASS
2	20	81	1	Safety ch...	2.0&20.0&81.0&1.0&Safety check: PASS
2	20	81	1	Safety ch...	2.0&20.0&81.0&1.0&Safety check: PASS
3	30	82	1	Manual t...	3.0&30.0&82.0&1.0&Manual test: PASS
1	10	80	99	AESStart:...	1.0&10.0&80.0&99.0&AESStart: PASS
1	10	80	99	AESStart:...	1.0&10.0&80.0&99.0&AESStart: PASS
2	20	81	1	Safety ch...	2.0&20.0&81.0&1.0&Safety check: PASS
2	20	81	1	Safety ch...	2.0&20.0&81.0&1.0&Safety check: PASS
2	20	81	1	Safety ch...	2.0&20.0&81.0&1.0&Safety check: PASS
3	30	82	1	Manual t...	3.0&30.0&82.0&1.0&Manual test: PASS
4	40	91	1	HV+EC t...	4.0&40.0&91.0&1.0&HV+EC test: PASS
4	40	91	1	HV+EC t...	4.0&40.0&91.0&1.0&HV+EC test: PASS
4	40	91	1	HV+EC t...	4.0&40.0&91.0&1.0&HV+EC test: PASS
4	40	91	1	HV+EC t...	4.0&40.0&91.0&1.0&HV+EC test: PASS

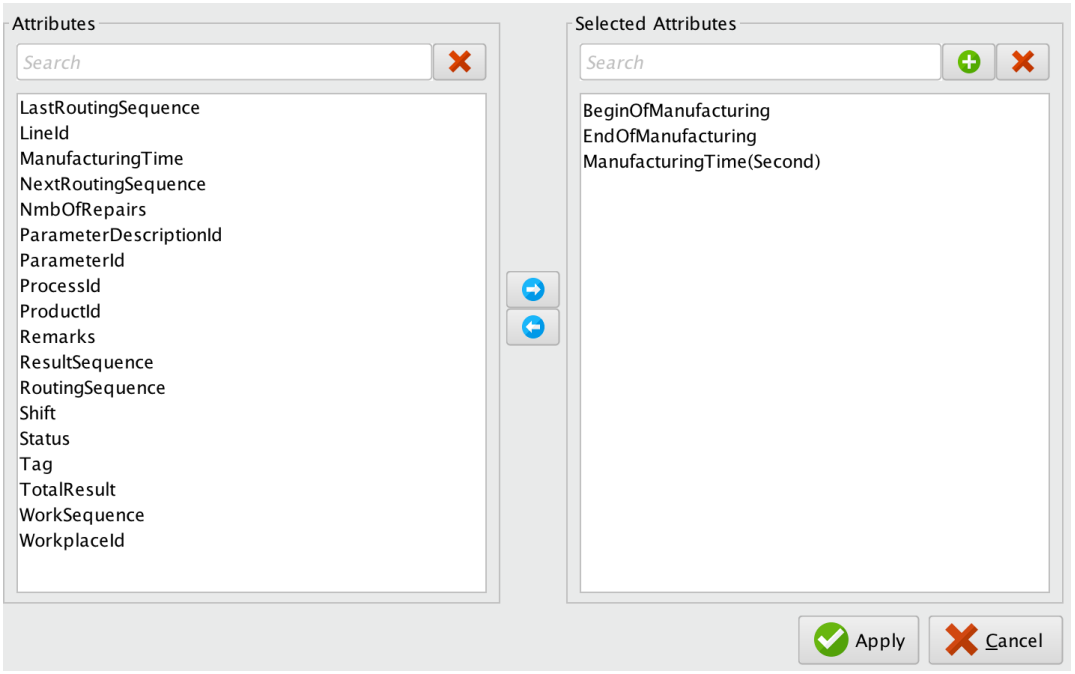
Anhang 7: Screenshot des Experimentprozesses der „Attribut-Extraktion“ aus den Attributen „BeginOfManufacturing“ und „EndOfManufacturing“



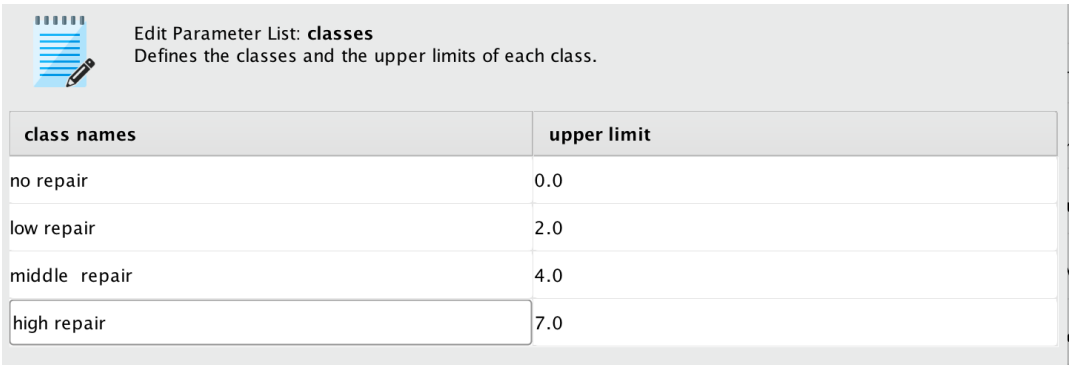
BeginOfManufacturing	EndOfManufacturing	ManufacturingTime
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM ...	929000



BeginOfManufacturing	EndOfManufacturing	ManufacturingTime	Manufactur...
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929
May 18, 2015 9:02:07 PM ...	May 18, 2015 9:17:36 PM CEST	929000	929

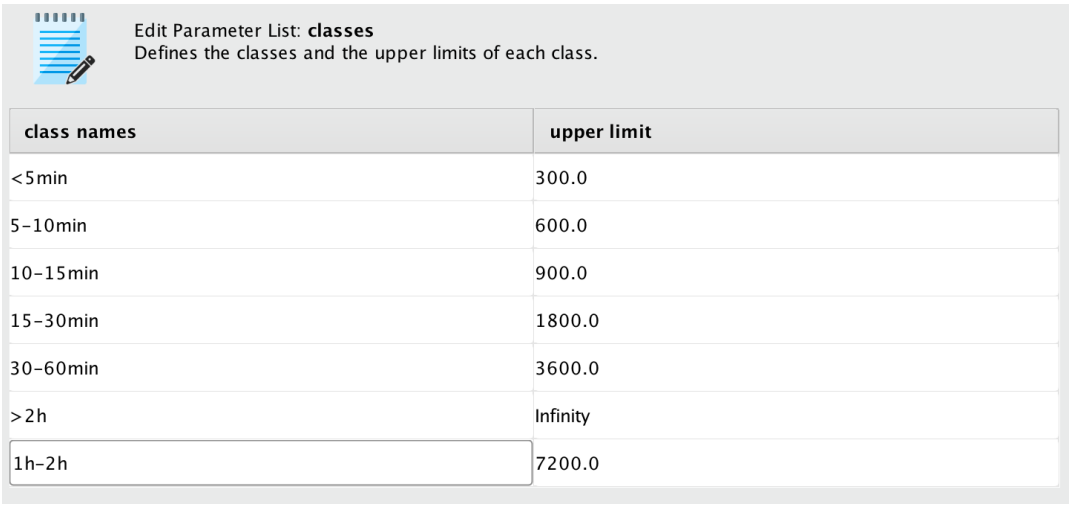


Anhang 8: Screenshot der Parametereinstellung und der Prozessergebnisse des Diskretisierungsprozesses der Attribute „NmbOfRepairs“ und „ManufacturingTime(Seclond)“



Edit Parameter List: **classes**
Defines the classes and the upper limits of each class.

class names	upper limit
no repair	0.0
low repair	2.0
middle repair	4.0
high repair	7.0



Edit Parameter List: **classes**
Defines the classes and the upper limits of each class.

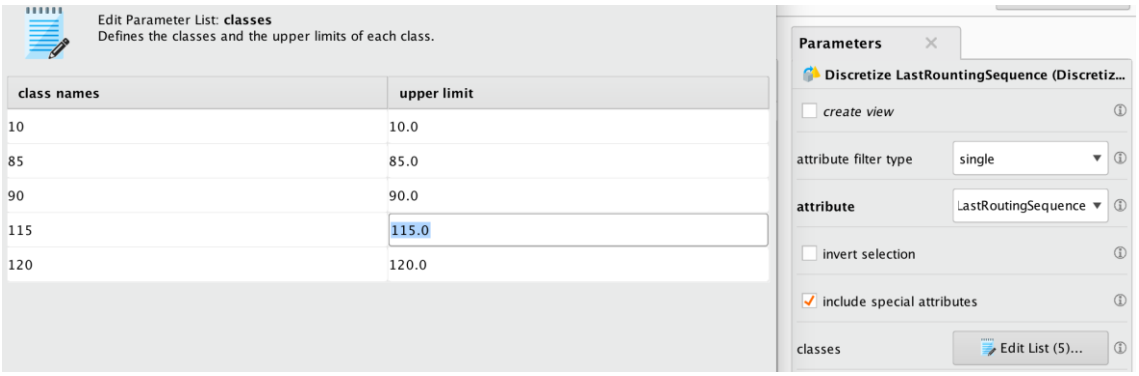
class names	upper limit
<5min	300.0
5-10min	600.0
10-15min	900.0
15-30min	1800.0
30-60min	3600.0
>2h	Infinity
1h-2h	7200.0

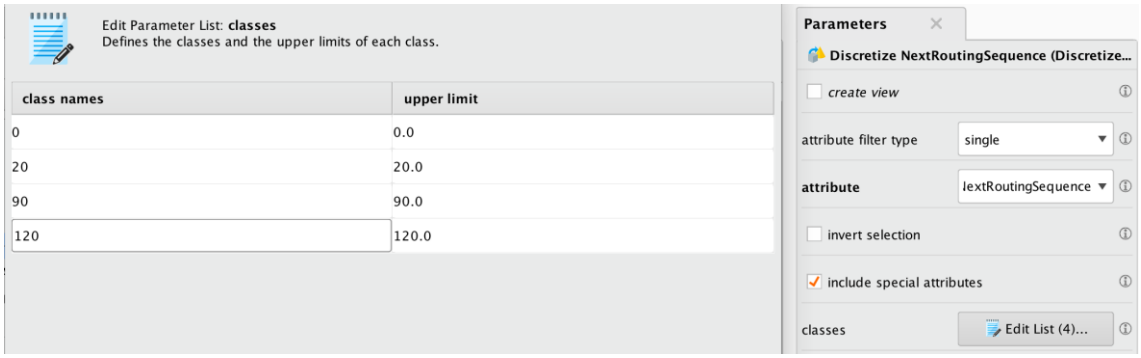
Row No.	Tag	ManufacturingTime(Second)	NmbOfRepairs
1	152123601...	15-30min	no repair
2	152123601...	15-30min	no repair
3	152123601...	15-30min	no repair
4	152123601...	15-30min	no repair
5	152123601...	15-30min	no repair
6	152123601...	15-30min	no repair
7	152123601...	15-30min	no repair
8	152123601...	15-30min	no repair
9	152123601...	15-30min	no repair
10	152123601...	15-30min	no repair

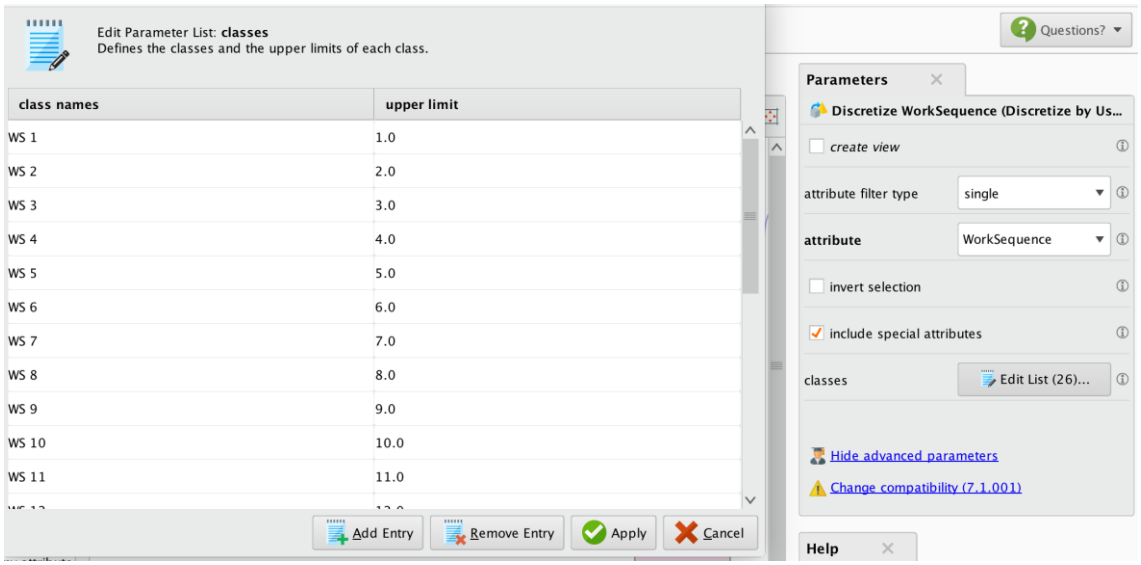
Anhang 9: Summe der fehlenden Werte von den Attributen (100.000 Datenzeilen)

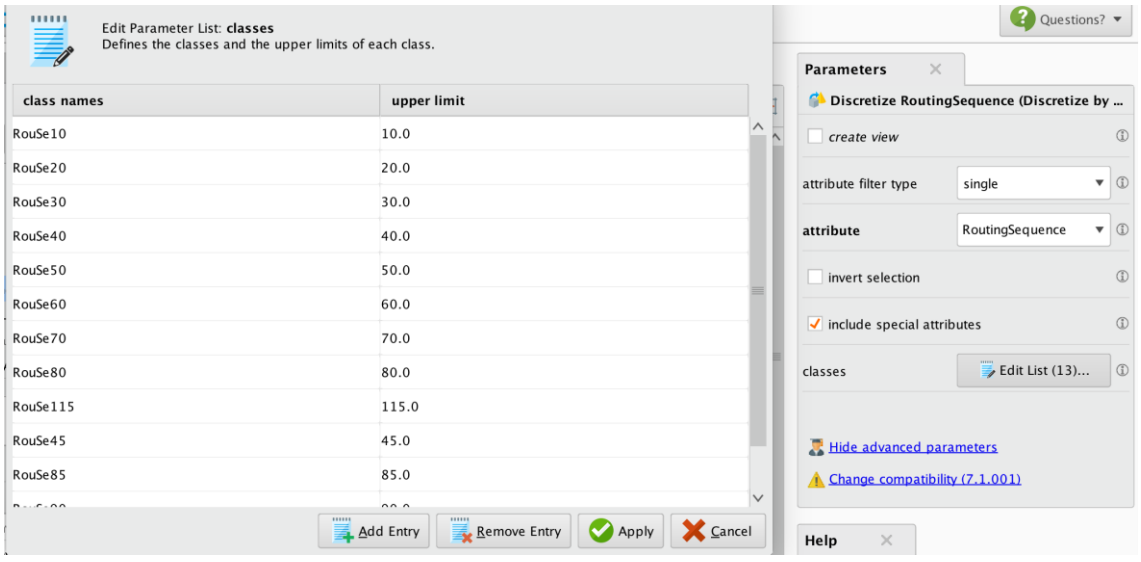
Attributname	Summe der fehlenden Werte	Attributname	Summe der fehlenden Werte
Guid	0	WorkSequence	11
LineId	11	RoutingSequence	11
Tag	183	WorkplaceId	11
ProductId	10	ProcessId	11
BeginOfManufacturing	10	Remarks	11
EndOfManufacturing	10	Status	11
SerialNumber1	11	ResultSequence	11
LastRoutingSequence	11	ParameterDescriptionId	11
NextRoutingSequence	11	ParameterId	11
TotalResult	10	NmbOfRepairs	11

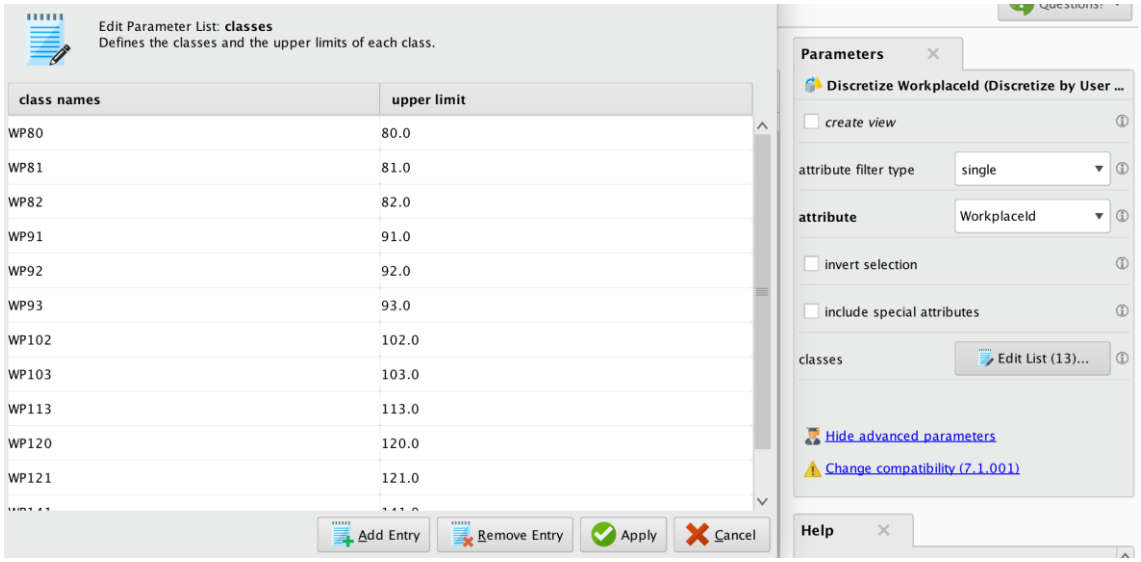
Anhang 10: Parametereinstellung der Diskretisierung zur Bereinigung der verrauschten Daten

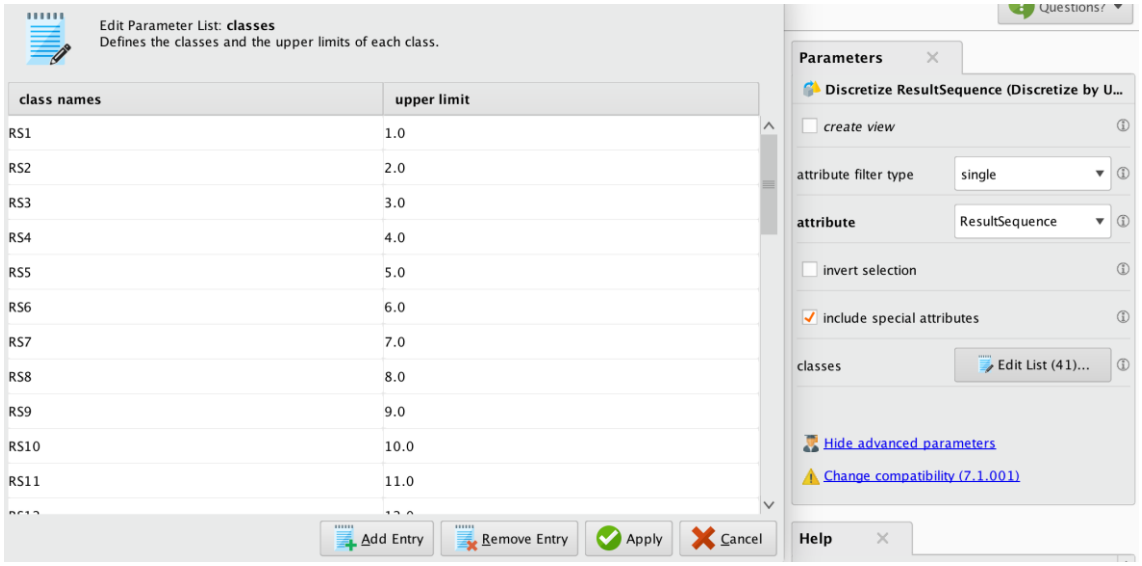



Edit Parameter List: classes
Defines the classes and the upper limits of each class.
class names
upper limit
0
0.0
20
20.0
90
90.0
120
120.0
Parameters
Discretize NextRoutingSequence (Discretize...
create view
attribute filter type
single
attribute
lextRoutingSequence
invert selection
include special attributes
classes
Edit List (4)...


Edit Parameter List: classes
Defines the classes and the upper limits of each class.
class names
upper limit
WS 1
1.0
WS 2
2.0
WS 3
3.0
WS 4
4.0
WS 5
5.0
WS 6
6.0
WS 7
7.0
WS 8
8.0
WS 9
9.0
WS 10
10.0
WS 11
11.0
Add Entry
Remove Entry
Apply
Cancel
Questions?
Parameters
Discretize WorkSequence (Discretize by Us...
create view
attribute filter type
single
attribute
WorkSequence
invert selection
include special attributes
classes
Edit List (26)...
Hide advanced parameters
Change compatibility (7.1.001)
Help

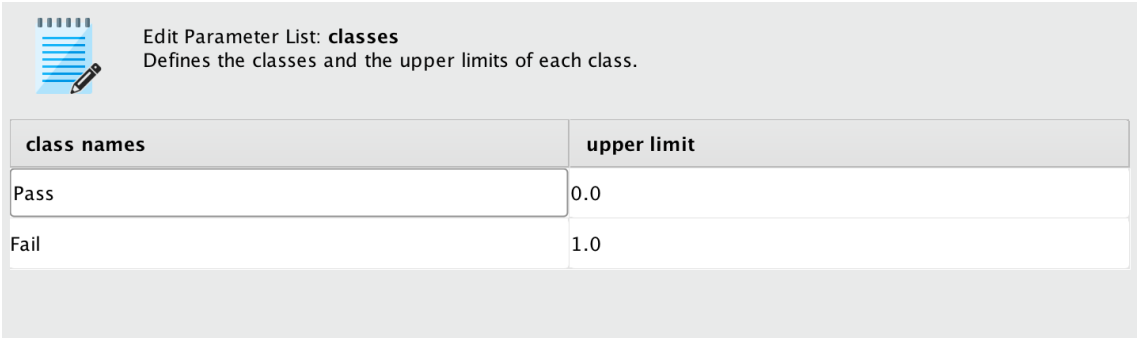

Edit Parameter List: classes
Defines the classes and the upper limits of each class.
class names
upper limit
RouSe10
10.0
RouSe20
20.0
RouSe30
30.0
RouSe40
40.0
RouSe50
50.0
RouSe60
60.0
RouSe70
70.0
RouSe80
80.0
RouSe115
115.0
RouSe45
45.0
RouSe85
85.0
Add Entry
Remove Entry
Apply
Cancel
Questions?
Parameters
Discretize RoutingSequence (Discretize by ...
create view
attribute filter type
single
attribute
RoutingSequence
invert selection
include special attributes
classes
Edit List (13)...
Hide advanced parameters
Change compatibility (7.1.001)
Help

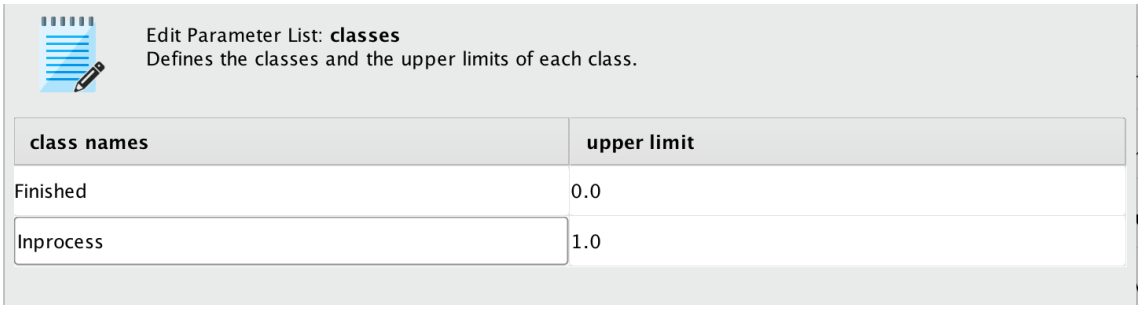

Edit Parameter List: classes
Defines the classes and the upper limits of each class.
class names
upper limit
WP80 80.0
WP81 81.0
WP82 82.0
WP91 91.0
WP92 92.0
WP93 93.0
WP102 102.0
WP103 103.0
WP113 113.0
WP120 120.0
WP121 121.0
Add Entry
Remove Entry
Apply
Cancel
Parameters
Discretize WorkplaceId (Discretize by User ...
create view
attribute filter type
single
attribute
WorkplaceId
invert selection
include special attributes
classes
Edit List (13)...
Hide advanced parameters
Change compatibility (7.1.001)
Help


Edit Parameter List: classes
Defines the classes and the upper limits of each class.
class names
upper limit
RS1 1.0
RS2 2.0
RS3 3.0
RS4 4.0
RS5 5.0
RS6 6.0
RS7 7.0
RS8 8.0
RS9 9.0
RS10 10.0
RS11 11.0
Add Entry
Remove Entry
Apply
Cancel
Parameters
Discretize ResultSequence (Discretize by U...
create view
attribute filter type
single
attribute
ResultSequence
invert selection
include special attributes
classes
Edit List (41)...
Hide advanced parameters
Change compatibility (7.1.001)
Help

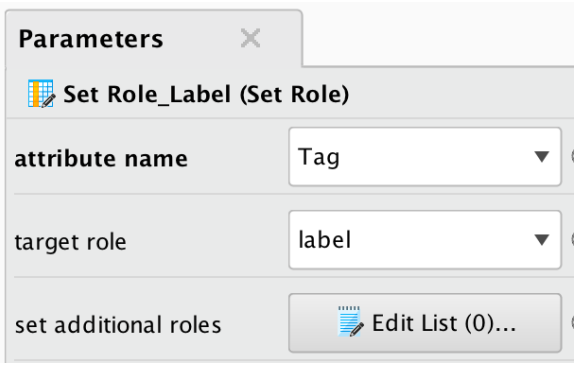

Edit Parameter List: classes
Defines the classes and the upper limits of each class.
class names
upper limit
Process1 1.0
Process99 99.0
Parameters
Discretize_ProcessId (Discretize by User Sp...
create view
attribute filter type
single
attribute
ProcessId
invert selection
include special attributes
classes
Edit List (2)...

Anhang 11: Parametereinstellung des Operators „Discretize“ vom Modellprozesses „Transformation des Datentyps von „binominal“ zum Datentyp „polynominal“

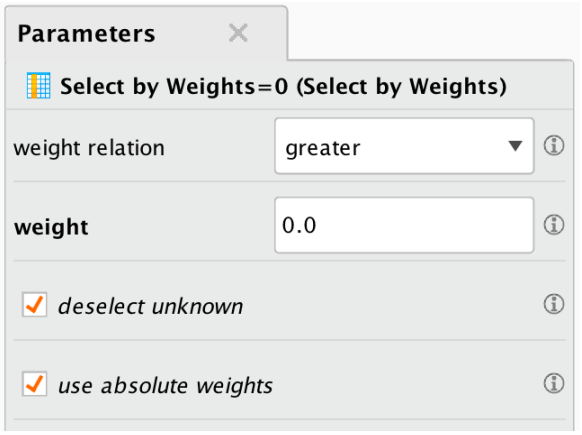


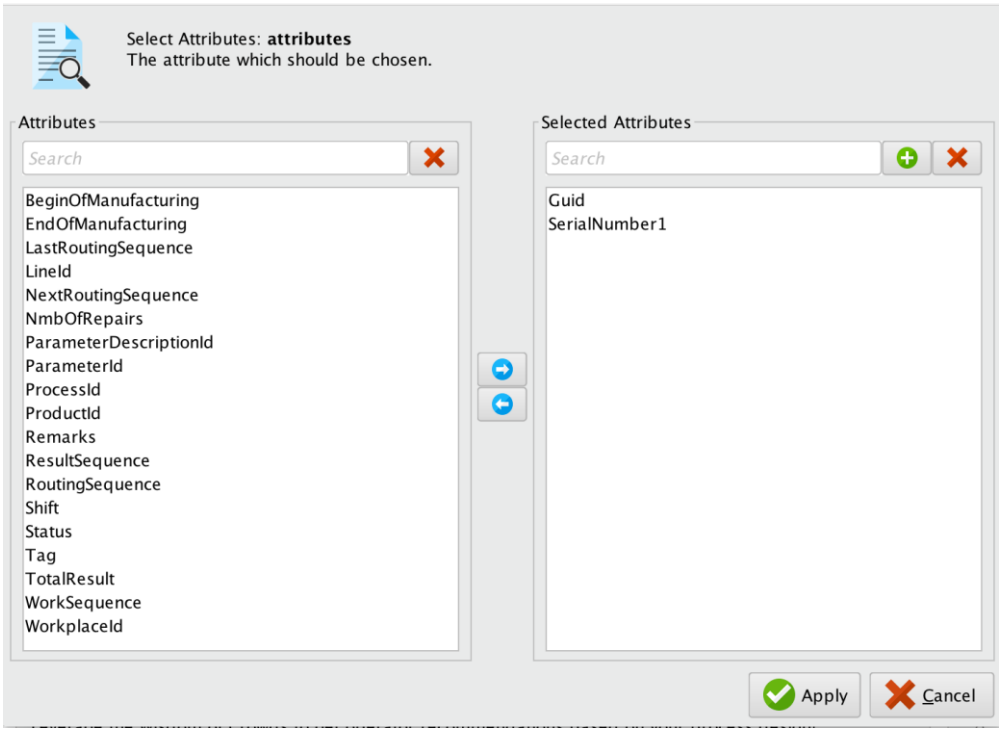


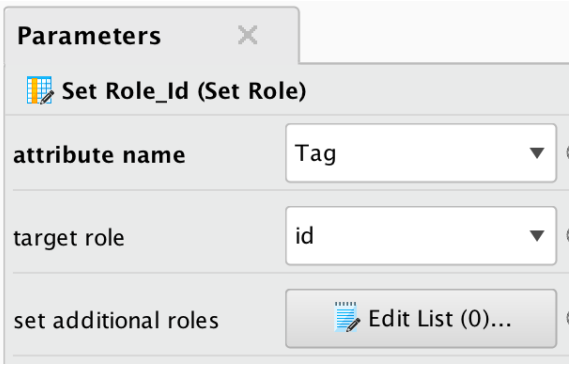
Anhang 12: Parametereinstellung und Zwischenergebnisse des Experimentprozesses „Chi Squared Statistik“



Remarks	0.001
TotalResult	0.001
WorkSequence	0.001
LineId	0.002
NmbOfRepairs	0.003
ProductId	0.004
ParameterId	0.005
SerialNumber1	1
Guid	1
BeginOfManufacturing	1
EndOfManufacturing	1







Anhang 13: Vergleichstabellen der „High repair“-Analyse

„High repair“-Analyse des Attributs „ManufacturingTime“

	HRDT	More7RDT	HDT
<5 min	0	0	12,4%
5-10 min	0	0	39,7%
10-15 min	0	0	28,3%
15-30 min	0	0	10,9%
30-60 min	0	0	3,1%
1-2 h	3,8%	0	3,4%
>2 h	96,2%	100%	2,2%

„High repair“-Analyse des Attributs „TotalResult“

	HRDT	More7RDT	HDT
PASS	92,4%	96,6%	99%
FAIL	7,6%	3,4%	1%

High repair“-Analyse des Attributs „LineId“

	HRDT	More7RDT	HDT
line 2	28%	21,7%	35,6%
line 3	36,4%	38%	30,3%
line 4	35,7%	40,3%	34,1%

„High repair“-Analyse des Attributs „ParameterDescriptionId“(1)

	HRDT	HDT
PDES00000164	7,4%	8,2%
PDES00000179	7,4%	8,2%
PDES0000092, 93, 95, 97, 98, 160, 167, 175	1,7%	Durchschnittlich 1,1%
PDES0000040-46, 48, 51, 91, 143, 150, 158, 173	1,6%	Durchschnittlich 1,1%
PDES00000028, 50, 165, 171	1,5%	Durchschnittlich 1,1%
PDES0000087, 137	1,4%	137: 2,3%, 87: 1,1%
PDES00000086, 88, 106, 148	1,3%	Durchschnittlich 1,1%
PDES00000110-114 144, 161, 168, 176	1,2%	Durchschnittlich 1,1%
Übrige Id-Nummern	≤ 1,1%	Durchschnittlich 1,1%

„High repair“-Analyse des Attributs „ProductId“(2)

	More7RDT	HDT
PDES00000164	7,2%	8,2%
PDES00000179	7,2%	8,2%
PDES0000092, 93, 95, 98, 160, 175	2%	Durchschnittlich 1,1%
PDES00000097, 167, 143	1,9%	Durchschnittlich 1,1%
PDES0000091, 150	1,7%	Durchschnittlich 1,1%
PDES00000040-46, 48, 51, 158, 173	1,5%	Durchschnittlich 1,1%
PDES00000148, 106, 110-112, 114, 161, 176	1,3%	Durchschnittlich 1,1%
PDES00000113, 144, 168	1,2%	Durchschnittlich 1,1%
Übrige Id-Nummern	≤ 1,1%	Durchschnittlich 1,1%

„High repair“-Analyse des Attributs „ProductId“

	HRDT	More7RDT	HDT
PROD00000006	32,3%	29,1%	39,5%
PROD00000009	26,3%	29,8%	8,8%
PROD00000010	24,1%	22,7%	20,8%
PROD00000011	14,7%	13,2%	2,6%
PROD00000012	2%	4%	27,6%
PROD00000014	0,7%	1,2%	0,7%

„High repair“-Analyse des Attributs „ResultSequence“

	HRDT	More7RDT	HDT		HRDT	More7RDT	HDT
RS1	8,1%	7,5%	9,3%	RS12	5,2%	5,3%	4,5%
RS2	6,9%	6,6%	8%	RS13	5,1%	5,1%	4,7%
RS3	6,9%	6,6%	8%	RS14	3,9%	3,9%	3,4%
RS4	6,9%	6,6%	8%	RS15	3%	3,2%	2,3%
RS5	6,4%	6,3%	6,9%	RS16	2,8%	3%	2,2%
RS6	6,4%	6,3%	6,9%	RS17	1,8%	1,9%	1,2%
RS7	6,3%	6,3%	6,9%	RS18	1,8%	1,9%	1,2%
RS8	6,3%	6,2%	6,7%	RS19-23	0,4%	0,5%	<0,1%
RS9	6,3%	6,1%	6,8%	RS24-26	0,3%	0,4%	<0,1%
RS10	6,2%	6,2%	6,9%	RS27-28	0,2%	0,3%	<0,1%
RS11	5,6%	5,5%	5,5%	RS29-36	0,1%	0,15%	<0,1%

„High repair“-Analyse des Attributs „RoutingSequence“

	HRDT	More7RDT	HDT
RouSe 10	28%	28,9%	4,5%
RouSe 20	25,9%	29,4%	21,2%
RouSe 30	15,6%	15,8%	15,9%
RouSe 40	15,5%	14,6%	18,6%
RouSe 50	5,7%	4,4%	14,9%
RouSe 60	6,4%	4,8%	11,3%
RouSe 70	1,9%	1,4%	12,4%
RouSe 80	0,5%	0,4%	1,1%
RouSe 115	0,5%	0,4%	0,1%

„High repair“-Analyse des Attributs „WorkSequence“

	HRDT	More7RDT	HDT		HRDT	More7RDT	HDT
WS 1	10,9%	29,6%	4,5%	WS 14	4,9%	3,7%	0,3%
WS 2	7,7%	6,1%	19,9%	WS 15	3,9%	2,2%	0,1%
WS 3	4,5%	3,6%	15,1%	WS 16	4,8%	3,6%	0,1%
WS 4	4,9%	4%	17,3%	WS 17	2,7%	2%	<0,1%
WS 5	2,6%	2%	0,8%	WS 18	4,3%	2,7%	0,1%
WS 6	4,6%	3,1%	14,4%	WS 19	3,8%	3,1%	≤ 1%
WS 7	3,2%	2,3%	11,3%	WS 20	3,1%	2,4%	≤ 0,1%
WS 8	2,6%	2,4%	11,7%	WS 21	3,2%	3,4%	≤ 0,1%
WS 9	3,7%	2,5%	1,9%	WS 22	2,6%	2,5%	≤ 0,1%
WS 10	3,2%	2,4%	0,8%	WS 23	2,5%	2,8%	≤ 0,1%
WS 11	4,7%	3,7%	0,9%	WS 24	2,1%	2,2%	≤ 0,1%
WS 12	2,5%	2,5%	0,4%	WS 25	1,8%	2,3%	≤ 0,1%
WS 13	3,1%	2,2%	0,4%	WS 26	2,1%	2,3%	≤ 0,1%

„High repair“-Analyse des Attributs „WorkplaceId“

	HRDT	More7 RDT	HDT		HRDT	More7 RDT	HDT
WP80	1,9%	1,4%	4,5%	WP93	15,6%	15,8%	14,9%
WP81	28%	28,9%	21,2%	WP102	6,4%	4,8%	11.3%
WP82	15,5%	14,6%	15,9%	WP103	5,7%	4,4%	12,3%
WP91	25,9%	29,4%	18,6%	WP113	1%	0,7%	1,2%

„High repair“-Analyse des Attributs „Remarks“

	HRDT	More7RDT	HDT
Safety check: PASS	20,3%	20,4%	20,5%
HV+EC test: PASS	19,2%	19,1%	18,5%
Manual test: PASS	12%	12,1%	15,8%
Calibration: PASS	8,1%	6,4%	14,6%
Flashing 1: PASS	5,1%	3,9%	11,3%
Flashing 2: PASS	5,1%	3,9%	12,2%
AESStart: PASS	1,7%	1,3%	4,5%
Marking: PASS	0,5%	0,4%	1,1%
Safety check: FAIL	7,7%	9,8%	0,7%
Calibration: FAIL	7,5%	9,5%	0,2%
Manual test: FAIL	3,4%	2,5%	0,2%
HV+EC test: FAIL	6,8%	10,3%	0,2%
Flashing 1: FAIL	1,3%	0,9%	0,1%
Flashing 2: FAIL	0,7%	0,6%	0,1%
Laser (9999999999IO-Teile): PASS	0,5%	0,4%	0,1%
AESStart: FAIL	0,2%	0,1%	<0,1%

„High repair“-Analyse des Attributs „ProcessId“

	HRDT	More7RDT	HDT
Process 1	98,1%	98,6%	95,5%
Process 99	1,9%	1,4%	4,5%

„High repair“-Analyse von „Aggregierten Attributen“

Aggregierte Attributwerte	HRDT	More7RDT
WS 2& RouSe20 & WP81 & Process1 & Safety check: PASS	5,5%	4,3%
WS 3 & RouSe30 & WP82 & Process1 & Manual test: PASS	3,1%	2,8%
WS 4 & RouSe40 & WP91 & Process1 & HV+EC test: PASS	2,9%	2,3%
WS 14 & RouSe20 & WP81 & Process1 & Safety check: PASS	2,3%	1,4%
WS 16 & RouSe40 & WP91 & Process1 & HV+EC test: PASS	2,3%	1,3%
WS 2 & RouSe20 & WP81 & Process1 & Safety check: FAIL	2,2%	1,8%
WS 9 & RouSe40 & WP91 & Process1 & HV+EC test: PASS	2%	1,5%
WS 6 & RouSe50 & WP93 & Process1 & Calibration: FAIL	2%	1,5%
WS 1 & RouSe70 & WP103 & Process1 & Flashing 2: PASS	1,8%	2,8%
WS 1 & RouSe10 & WP80 & Process99 & AESStart: PASS	1,7%	1,3%
WS 1 & RouSe50 & WP93 & Process1 & Calibration: PASS	1,7%	3,8%

WS 1 & RouSe40 & WP91 & Process1 & HV+EC test: PASS	1,6%	6,4%
WS 11 & RouSe50 & WP93 & Process1 & Calibration: FAIL	1,6%	1,2%
WS 15 & RouSe30 & WP82 & Process1 & Manual test: PASS	1,6%	
WS 5 & RouSe20 & WP81 & Process1 & Safety check: FAIL	1,5%	1,5%
WS 1 & RouSe60 & WP102 & Process1 & Flashing 1: PASS	1,5%	2,6%
WS 3 & RouSe30 & WP82 & Process1 & Manual test: FAIL	1,2%	
WS 6 & RouSe20 & WP81 & Process1 & Safety check: PASS	1,2%	
WS 8 & RouSe20 & WP81 & Process1 & Safety check: FAIL	1,1%	1,4%
WS 4 & RouSe40 & WP91 & Process1 & HV+EC test: FAIL	1,1%	1,3%
WS 18 & RouSe50 & WP93 & Process1 & Calibration: PASS	1,1%	
WS 14 & RouSe40 & WP91 & Process1 & HV+EC test: PASS	1,1%	
WS 7 & RouSe40 & WP91 & Process1 & HV+EC test: FAIL	1%	1,1%
WS 1 & RouSe20 & WP81 & Process1 & Safety check: PASS		4,2%
WS 1 & RouSe30 & WP82 & Process1 & Manual test: PASS		3,9%
WS 1 & RouSe50 & WP93 & Process1 & Calibration: FAIL		2,2%
WS 11 & RouSe20 & WP81 & Process1 & Safety check: FAIL		1,2%
WS 14 & RouSe20 & WP81 & Process1 & Safety check: FAIL		1%

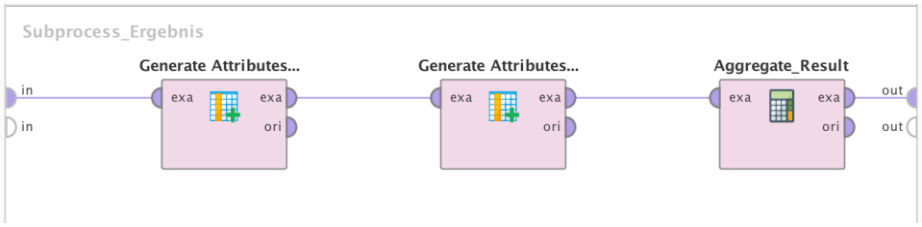
Anhang 14: Die Ergebnisse von der Aggregation der Attributwerte

No repair	89,7%	ParaDesId_137, 164, 179	18,6%	RoutingSequence_mFre.	38,5%
low repair	9%	ParaDesId_24-70	21,0%	RoutingSequence_lFre.	5,8%
middle repair	1%	ParaDesId_74-124	20,9%	WorkplaceId_hFre.	55,7%
high repair	0,2%	ParaDesId_125-160	20%	WorkplaceId_mFre.	38,5%
repair>7	0,1%	ParaDesId_161-180	19,5%	WorkplaceId_lFre.	5,8%
Process 1	95,5%	ProductId_hFre	67,1%	ResultSequence_hFre.	80,0%
Process 99	4,5%	ProductId_mFre	29,7%	ResultSequence_mFre.	19,5%
line 2	35,6 %	ProductId_lFre	3,2%	ResultSequence_lFre.	0,5%
line 3	30,3 %	WorkSequence_hFre.	89,7%	ManufacturingTime (Second)_hFre	67,8%
line 4	34,1 %	WorkSequence_mFre.	6,4%	ManufacturingTime (Second)_mFre	23,4%
PASS	99%	WorkSequence_lFre.	3,9%	ManufacturingTime (Second)_lFre	8,8%
FAIL	1%	RoutingSequence_hFre.	55,7%		

Anhang 15: Anzahl der maximalen Durchläufe und der maximalen Optimierungsschritte des k-Means-Algorithmus

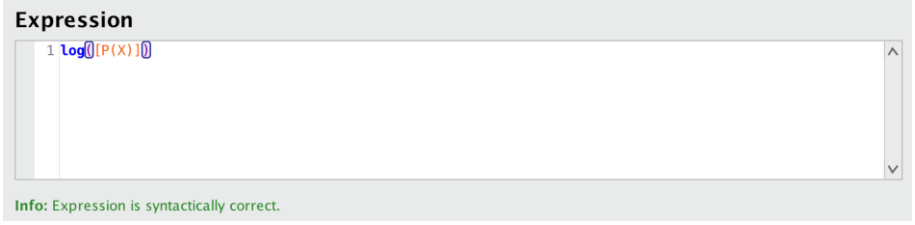
HDT mit Top 100.000 Datenzeilen			HDT mit Last 100.000 Datenzeilen			FDT			HRDT			More7RDT		
K-Wert	Max. Durchläufe	Max. Optimierungsschritte	K-Wert	Max. Durchläufe	Max. Optimierungsschritte	K-Wert	Max. Durchläufe	Max. Optimierungsschritte	K-Wert	Max. Durchläufe	Max. Optimierungsschritte	K-Wert	Max. Durchläufe	Max. Optimierungsschritte
2	7	8	2	5	4				2	62	4	2	18	4
3	5	6	3	40	20	3	5	6	3	175	10	3	15	3
4	9	8	5	3	7	4	17	9						

Anhang 16: Experimentprozess vom EM-Algorithmus

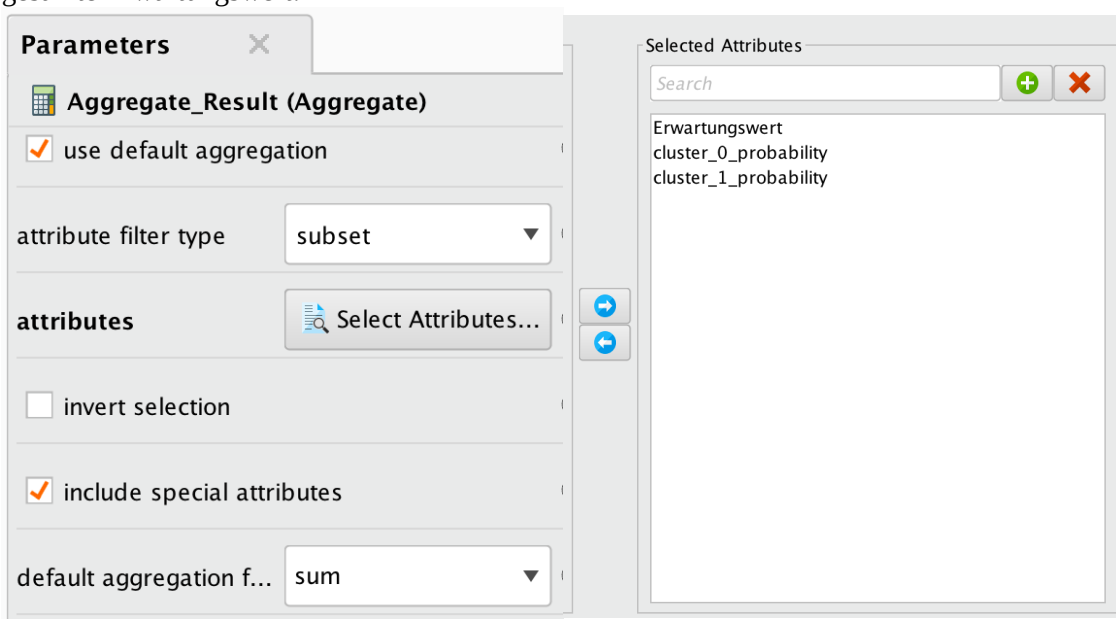


Berechnungsfunktion von *p(x)*:

Berechnungsfunktion von *E*:



Berechnung von den gesamten Wahrscheinlichkeiten jedes Cluster ohne Gewichtung und der gesamte Erwartungswert:



Anhang 17: Anzahl der maximalen Durchläufe und maximale Optimierungsschrittes des EM-Algorithmus

HDT_AESBig mit Top 100.000 Datenzeilen			HDT_AESBig mit Last 100.000 Datenzeilen			FDT			HRDT			More7DT		
K-Wert	Max. Durchläufe	Max. Optimierungsschritte	K-Wert	Max. Durchläufe	Max. Optimierungsschritte	K-Wert	Max. Durchläufe	Max. Optimierungsschritte	K-Wert	Max. Durchläufe	Max. Optimierungsschritte	K-Wert	Max. Durchläufe	Max. Optimierungsschritte
2	4	1	2	4	1				2	3	1	2	4	1
3	4	1	3	4	1	3	n. b.	n. b.	3	2	1	3	2	1
4	4	1	5	4	1	4	3	1						

Anhang 18: Ergebnisse der Davies-Bouldin-Index-Methode

	HDT mit Top 100.000 Daten-zeilen	HDT mit Last 100.000 Daten-zeilen	FDT	HRDT	More7DT
K=2	2,171	1,362		3,013	2,768
K=3	1,834	2,314	2,122	2,496	2,885
K=4	2,420		2,179		
K=5		2,027			

Anhang 19: Ergebnisse der Wahrscheinlichkeitsmaße-Methode

	HDT mit Top 100.000 Datenzeile	HDT mit Last 100.000 Datenzeile	FDT	HRDT	More7DT
K=2	-3938,746	-3101,903		-2726,424	-16214,572
K=3	-4131,116	-7523,228	n. b.	-11734,332	-19101,996
K=4	-15603,878		-14114,511		
K=5		-27006,782			

Anhang 20: Clusteranalyse-Ergebnisse vom k-Means-Algorithmus von der Tabellenform

	Cluster_PASS		Cluster_low repair		Cluster_middle repair	
NmbOf-Repairs	no repair: low re-pair:	99,6% (89,7%) 0,4% (9%)	low repair: mid. repair: high repair: repair>7:	83,5% (9%) 12,0% (1%) 2,6% (0,2%) 1,9% (0,1%)	mid. repair: high repair: repair>7:	99,1% (1%) 0,88% (0,2%) 0,02% (0,1%)
ProcessId	Process 1: Process 99:	95,3% 4,7%	Process 1:	100%	Process 1: Process 99:	94,4% (95,5%) 5,6% (4,5%)
LineId	Line 2: Line 3: Line 4:	37,2% (35,6%) 29,5% (30,3%) 33,3% (34,1%)	Line 2: Line 3: Line 4:	16,3% (35,6%) 51,3% (30,3%) 32,4% (34,1%)	Line 2: Line 3: Line 4:	39,4% (35,6%) 41,2% (30,3%) 19,4% (34,1%)
TotalRe-sult	PASS:	100%	PASS: FAIL:	86,17% 13,83%	FAIL: 100% (Daten aus der FDT)	

Parameter Descripti-onId	Id 137, 164, 179: Id 24-70: Id 74-124: Id 125-160: Id 161-180:	18,8% 20,9% 21,7% (20,9%) 19,0% (20%) 18,5% (19,5%)	Id 137, 164, 179: Id 24-70: Id 74-124: Id 125-160: Id 161-180:	15,5% (18,6%) 19,8% (21,0%) 24,1% (20,9%) 20,8% 19,8%	Id 137, 164, 179: Id 24-70: Id 74-124: Id 125-160: Id 161-180:	18,4% 39.1% (21,0%) 15,8% (20,9%) 12,8% (20%) 13,9% (19,5%)
ProductId	hFre: mFre: lFre:	67,2% 29,4% 3,4%	hFre: mFre: lFre:	48,9% (67,1%) 49,5% (29,7%) 1,6% (3,2%)	hFre: mFre:	81,4% (67,1%) 18,6% (29,7%)
Work Sequence	hFre: mFre: lFre:	94,2% (89,7%) 5,8% (6,4%) 0% (3,9%)	hFre: mFre: lFre:	39,0% (89,7%) 10,1% (6,4%) 50,9% (3,9%)	hFre: mFre: lFre:	55,0% (89,7%) 20,9% (6,4%) 24,1% (3,9%)
Routing Sequence	hFre: mFre: lFre:	55,2% 39,0% 5,8%	hFre: mFre: lFre:	55,5% 43,8% (38,5%) 0,7% (5,8%)	hFre: lFre:	91,8% (55,7%) 8,2% (5,8%)
Work-placeId	hFre: mFre: lFre:	55,2% 39% 5,8%	hFre: mFre: lFre:	55,5% 43,8% (38,5%) 0,7% (5,8%)	hFre: lFre:	91,8% (55,7%) 8,2% (5,8%)
ResultSe-quence	hFre: mFre: lFre:	80,1% 19,4% 0,5%	hFre: mFre: lFre:	78,1% (80,0%) 20,2% 1,7% (0,5%)	hFre: mFre: lFre:	69,6% (80,0%) 29,2% (19,5%) 1,2% (0,5%)
Manufac-turing Time	hFre: mFre: lFre:	75,6% (67,8%) 24,0% 0,4% (8,8%)	hFre: mFre: lFre:	1,1% (67,8%) 6,4% (23,4%) 92,5% (8,8%)	lFre:	100% (67,8%)

Anhang 21: Clusteranalyse-Ergebnisse vom EM-Algorithmus von der Tabellenform

	Cluster_PASS		Cluster_no repair		Cluster_low repair	
NmbOf Repairs	no repair: low repair: mid. repair: high repair: repair>7:	91,9% (89,7%) 7,1% (9%) 0,8% (1%) 0,1% (0,2%) 0,02% (0,1%)	no repair: low repair:	97,6% (89,7%) 2,4% (9%)	low repair: mid. repair: high repair: repair>7:	91% (9%) 7,4% (1%) 1,6% (0,2%) 0,1%
ProcessId	Process 1: Process 99:	95,4% 4,6%	Process 1:	100%	Process 1: Process 99:	95,2% 4,8%
LineId	Line 2: Line 3: Line 4:	34,9% (35,6%) 31,6% (30,3%) 33,4% (34,1%)	Line 2: Line 3: Line 4:	35,4% 30,8% 33,8%	Line 2: Line 3: Line 4:	33,6% (35,6%) 32,7% (30,3%) 33,7%
TotalResult	PASS: FAIL:	99,7% (99%) 0,3% (1%)	PASS:	100%	FAIL: 100% (Daten aus der FDT)	
Parameter Description Id	Die RH von allen Id-Nummer in diesem Cluster sind identisch zur RH der jeweiligen Id-Nummer in der HDT		Id 137: Id 164: Id 179: Id 27:	1,3% (2,3%) 3,8% (8,2%) 3,8% (8,2%) 0% (1,1%)	Id 137: Id 164: Id 179:	3,6% (2,3%) 8,7% (8,2%) 8,7% (8,2%)
			Die RH von allen übrigen Id-Nummer in diesem Cluster sind identisch zur RH der jeweiligen Id-Nummer in der HDT		Die RH von allen übrigen Id-Nummer in diesem Cluster sind identisch zur RH der jeweiligen Id-Nummer in der HDT	
ProductId	Id 06:	40,5%	Id 09:	7,2%	Id 10:	21,6%

	Id 09:	(39,5%) 7,3%	Id 12:	(8,8%) 28,7%	Id 11:	(20,8%) 1,9%
	Id 10:	(8,8%) 21,5%	Id 14:	(27,6%) 0,4%	Id 14:	(2,6%) 1,2%
	Id 11:	(20,8%) 2,3%		(0,7%)		(0,7%)
	Id 14:	(2,6%) 0,46%				
		(0,7%)				
	Id 12		Id 06, 10, 11		Id 06, 09, 12	
Work Sequence	WS 2:	20,2%	WS 2:	20,8%	WS 2:	1,2%
		(19,9%)		(19,9%)		(21,2%)
	WS 3:	15,4%	WS 3:	15,7%	WS 5:	15,8%
		(15,1%)		(15,1%)		(0,8%)
	WS 4:	17,6%	WS 4:	18,3%	WS 7:	1,1%
		(17,3%)		(17,3%)		(11,3%)
	WS 5:	0,5%	WS 5:	0,006%	WS 8:	15,5%
		(0,8%)		(0,8%)		(11,7%)
			WS 6:	16,7%	WS 9:	12,3%
	WS 10:	0,6%(0,8%)		(14,4%)		(1,9%)
		0,04%		13,2%	WS 10:	12,1%
		(0,1%)	WS 7:	(11,3%)		(0,8%)
	WS 16:	0,03%		14,0%	WS 11:	2,2%
	WS 18:	(0,1%)	WS 8:	(11,7%)		(0,9%)
				1,3%	WS 12:	1,4%
	WS 1, WS 6 – WS 9 WS 11 – WS 15, WS 19 – WS 23, WS 25 – WS 28		WS 9:	(1,9%)		(0,4%)
					WS 13:	1,2%
						(0,4%)
					WS 14:	0,6%
						(0,3%)
					WS 15:	0,4%
						(0,1%)
					WS 16:	0,3%
						(0,1%)
					WS 17:	0,2%
						(<0,1%)
					WS 18:	0,05%
						(0,1%)
					WS 1, 4, 6, 20, 21, 23,	
Routing Sequence	RouSe 10, 20, 30, 40, 50, 60, 70, 80, 115		RouSe 50:	16,6% (14,9%)	RouSe 20:	18,8% (21,2%)

		RouSe 60: RouSe 70:	12,7% (11,3%) 14,0% (12,4%)	RouSe 30: RouSe 40: RouSe 115:	17,3% (15,9%) 16 % (18,6%) 1,2% (0,1%)
		RouSe 20, 30, 40, 80		RouSe 10, 50, 60, 70, 80	
Work-placeId	WP 80, 81, 82, 91, 93, 102, 103, 113	WP 93: WP 102: WP 103:	16,6% (14,9%) 12,7% (11,3%) 14,0% (12,3%)	WP 81: WP 82: WP 91: WP 93: WP 113:	18,8% (21,2%) 17,3% (15,9%) 16% (18,6%) 15,7% (14,9%) 2,4% (1,2%)
		WP 81, 82, 91, 113		WP 80, 102, 103	
Result Sequence	RS 1 - RS 36	RS 4: RS 5: RS 6: RS 7: RS 9: RS 10: RS 12: RS 13: RS 14: RS 15:	6,5% (8%) 6,5% (6,9%) 7,8% (6,9%) 6,5% (6,9%) 7,7% (6,8%) 7,8% (6,9%) 3,8% (4,5%) 5,3% (4,7%) 3,9% (3,4%) 2,7% (2,3%)	RS 1: RS 2: RS 3: RS 4: RS 5: RS 6: RS 7: RS 9: RS 10: RS 11: RS 12: RS 13: RS 15: RS 16: RS 19 :	11% (9,3%) 8,6% (8%) 7,5% (8%) 7,3% (8%) 6,1% (6,9%) 7,4% (6,9%) 7,4% (6,9%) 6,2% (6,8%) 7,3% (6,9%) 5,9% (5,5%) 4,9% (4,5%) 3,9% (4,7%) 1,3% (2,3%) 1% (2,2%) 0,1%

					RS 20 :	(<0,1%) 0,1% (<0,1%)
			RS 1, 2, 3, 8 11, 16-34		RS 8, 14, 17, 18, 21-36	
Manufac-turing Time (Second)	10-15 min: 15-30 min: 30-60 min: 1-2h: >2h:	29,3% (28,3%) 12,2% (10,9%) 2,8% (3,1%) 1,9% (3,4%) 1,8% (2,2%)	5-10 min: 10-15 min: 15-30 min: 30-60 min: 1-2 h: > 2 h:	42,4% (39,7%) 30,9% (28,3%) 12,0% (10,9%) 1,2% (3,1%) 0,4% (3,4%) 0,2% (2,2%)	15-30 min: 1-2 h: > 2 h:	13,1% (10,9%) 58% (3,4%) 28,9% (2,2%)
	<5 min		<5 min			
Remarks	Flashing 2: FAIL: Manual test: FAIL: Safety check: FAIL: HV+EC test: FAIL:	0,01% (0,1%) 0,014% (0,2%) 0,5% (0,7%) 0,1% (0,2%)	Calibration: PASS: Flashing 1: PASS: Flashing 2: PASS: Marking: PASS:	16,6% (14,6%) 12,7% (11,3%) 14,0% (12,2%) 1,3% (1,1%)	AESStart: FAIL: Calibration: FAIL: Calibration: PASS: Flashing 1: FAIL: HV+EC test: PASS: Laser (9999999999IO-Teile): PASS: Manual test: PASS: Safety check: FAIL: Safety check:	4,8% (<0,1%) 0,1% (0,2%) 15,6% (14,6%) 0,2% (0,1%) 15,9% (18,5%) 1,2% (0,1%) 17,1% (15,8%) 1,1% (0,7%) 17,7%
	AESStart: PASS, AESStart: FAIL, Calibration: FAIL, Calibration: PASS, Flashing 1: PASS, Flashing 2: PASS, HV+EC test: PASS, Laser (9999999999IO-Teile): PASS, Manual test: PASS, Marking: PASS, Safety check: PASS					

				PASS:	(20,4%)
		HV+EC test: PASS, Manual test: PASS, Safety check: PASS		Flashing 1: PASS, Flashing 2: PASS, HV+EC test: FAIL, Manual test: FAIL, Marking: PASS	

Eidesstattliche Versicherung

_____ _____

Name, Vorname Matr.-Nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit/ Masterarbeit* mit dem Titel

_____

_____

_____

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

_____ _____

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfls. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

_____ _____

Ort, Datum Unterschrift