Modellentwicklung für die Ableitung von Typregionen der Energieversorgung

Freie wissenschaftliche Arbeit zur Erlangung
des Grades eines Bachelor of Science

von

Julian Endres
Matrikel-Nummer: 353273

Technische Universität Berlin
Fakultät VII - Institut für Technologie und Management
Fachgebiet Energie- und Ressourcenmanagement

Erster Gutachter: Prof. Dr. J. Müller-Kirchenbauer

Zweiter Gutachter: M.Sc. Benjamin Grosse

Datum der Abgabe: 13.12.2019

Eidesstattliche Erklärung

Von:	Endres, Julian
Matrikel-Nummer:	353273

Eidesstattliche Erklärung Ich erkläre hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit/Studienarbeit/Diplomarbeit/Masterarbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift)

Zusammenfassung

Die flächendeckende Integration Erneuerbarer Energien ist eine gesellschaftliche Herausforderung. Zahlreiche Studien und Forschungsprojekte begleiten den Prozess der Transformation und die Auswirkung der Energiewende auf bestehende Infrastrukturen. Jedoch sind viele Erkenntnisse durch eine begrenzte räumliche Auflösung eingeschränkt. Die Ableitung von repräsentativen Typregionen ist ein möglicher Lösungsansatz um Regionen zu identifizieren, die exemplarisch für eine größere Gruppe stehen.

In dieser Abschlussarbeit wurde ein Modell zur Identifikation von Typregionen (ModITy) auf NUTS-3 Ebene entwickelt. Im Wesentlichen ist ein einfach zu handhabendes Werkzeug zum Erstellen von robusten Clusterzuordnungen entstanden. Hierbei ist die Ergebnisgüte unabhängig von dem thematischen Schwerpunkt der Eingangsdaten. Das Modell ist somit auch außerhalb der Energiewirtschaft nutzbar. Unter Verwendung von Methoden des maschinellen Lernens werden die Daten mit mathematischen Algorithmen auf Gemeinsamkeiten untersucht und Cluster mit Regionen ähnlicher Eigenschaften gebildet. Die Zuordnung der Cluster erfolgt mit Hilfe des K-Means Algorithmus. Der Aufbau ist jedoch modular und kann somit um zusätzliche Verfahren erweitert werden. Um die Clustergüte zu optimieren werden interne Validierungsindizes verwendet. Anhand der Clusterschwerpunkte lassen sich Typregionen ableiten. Diese stehen repräsentativ für ihr Cluster. Die Regionen innerhalb eines Clusters weißen sich durch eine Nähe zueinander aus. Die einzelnen Cluster grenzen sich jedoch bestmöglich voneinander ab. Eine Zugehörigkeit zu einem Cluster deutet somit darauf hin, dass Forschungsergebnisse anderer Clustermitglieder übertragbar sein können. Die Ergebnisse dienen als Grundlage, um regionale Forschungsergebnisse innerhalb eines Clusters zu übertragen. Außerdem können an den Typregionen Forschungsfragen exemplarisch für das ganze Cluster untersucht werden.

In der Arbeit werden die angewendeten Methoden erklärt, ihre Implementierung beschrieben und das Modell exemplarisch in zwei Varianten angewendet. Die Ergebnisse werden diskutiert und dabei die Grenzen des Modells aufgezeigt. Abschließend wird ein Ausblick auf sinnvolle Erweiterungen gegeben.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

4ÜNB 4 Übertragungsnetzbetreiber: 50Hertz, Amprion, Tennet, TransnetBW

AIRE Anforderungen an die Infrastrukturen im Rahmen der Energiewende, Projekt

BBSR Bau, Stadt- und Raumforschung

BEV Battery Electric Vehicle, Batterieelektrisches Auto

BHKW Blockheizkraftwerk

BIP Bruttoinlandsprodukt

BMWi Bundesministerium für Wirtschaft und Energie

Comp Compactness eng. Kompaktheit

EE Erneuerbare Energien

EEG Erneuerbare Energien Gesetz

FCEV Fuel Cell Electric Vehicle, Brennstoffzellenauto

Sep Seperation eng. Trennung

1 Einleitung

1.1 Hintergrund und Motivation

Der Klimawandel ist eine globale Herausforderung, derer sich die vereinten Nationen mit dem Pariser Übereinkommen angenommen haben. Trotz der Selbstverpflichtungen zu CO_2-Reduktionen der einzelnen Mitgliedstaaten ist nur vereinzelt von Erfolgen zu sprechen. Viele Nationen kämpfen mit den selbst gesteckten Zielen. Denn mit dem Prozess der Transformation sind enorme gesellschaftliche und wirtschaftliche Herausforderungen verbunden. Deutschland wird seine Klimaziele für 2020 voraussichtlich um fünf Jahre verfehlen. Um das 40 % Reduktionsziel in absehbarer Zeit zu erreichen, muss der Kohleausstieg zügig umgesetzt bzw. sogar beschleunigt werden. Dies geht Hand in Hand mit dem zunehmenden Ausbau der Erneuerbaren Energien. Der jedoch momentan zum erliegen gekommen ist.[1] In Regionen mit viel Windkraftanlagen und entlang der Nord-Süd Trasse kommt es durch Akzeptanzproblemen zu Verzögerungen im Ausbau. Zahlreiche Studien und Forschungsprojekte begleiten deshalb den Prozess der Transformation. Jedoch sind viele Modelle zu komplex und schwer zu verstehen. Oder Erkenntnisse sind durch eine begrenzte räumliche Auflösung eingeschränkt und nicht auf andere Regionen übertragbar.[2] Vergleicht man Regionen in ihrer gesamten Vielfalt, fällt es schwer, Gleichartigkeiten hinsichtlich einer Fragestellung zu identifizieren. Die Ableitung von repräsentativen Typregionen ist ein möglicher Lösungsansatz um Regionen zu identifizieren, die exemplarisch für eine größere Gruppe stehen. Hierbei werden Regionen auf ihre Gemeinsamkeit hin untersucht und zu Clustern zusammengefasst. Hierdurch kann sich die Analyse auf wenige Typregionen pro Cluster vereinfachen, obwohl die Aussage und Darstellung für alle im statistischen Rahmen erhalten bleibt. Die Ergebnisse dienen somit als Grundlage zur Übertragbarkeit von regionalen Forschungsergebnissen aber auch für bei der Auswahl ähnlicher Regionen für weitere Analysen bzgl. einer bestimmten Forschungsfrage. Zusätzlich kann man die Zugehörigkeiten nutzen, um den Informationsaustausch innerhalb der identifizierten Cluster durch das Aufzeigen von Gemeinsamkeiten zu befördern und Partnerschaften anzuregen.

[1] Greenpeace, 2019.
[2] Keles u. a., 2017.

1.2 Zielsetzung

In dieser Arbeit wird ein Modell zur Ableitung von Typregionen entwickelt. Ein erstes Modell entstand im Rahmen des Seminars ER-SAM „Simulation, Analyse und Methoden“ im Wintersemester 2019 am Fachgebiet Energie- und Ressourcenmanagement. Die Studierenden Tom Sudhaus, Theodor Schönfisch und Anne Droldner fokussierten sich hinsichtlich der Daten auf das Bruttoinlandsprodukt (BIP) sowie Gasverbräuche in Industrie, Gewerbe, Handel, Dienstleistung und privaten Haushalten. Die Clusteranalyse erfolgte innerhalb der siedlungsstrukturellen Einteilung des Bundesinstitutes für Bau, Stadt-, und Raumforschung (BBSR). Insgesamt wurden die vier BBSR-Cluster jeweils in zwei Untercluster geteilt. Diese vordefinierte Raumeinteilung soll durch Rohdaten ersetzt, mit zusätzlichen Einflussgrößen und qualitativen Prüfmethoden erweitert werden. Der Fokus in dieser Arbeit liegt auf der Modellentwicklung. Es soll ein robustes Modell entstehen, das zuverlässig mit unterschiedlichen Datensätzen arbeitet. Ein thematischer Bezug der Eingangsdaten zur Energiewirtschaft ist nicht notwendig, obwohl das Modell hierfür entwickelt wird. Bei der Suche nach „noch unentdeckten“ Strukturen werden sogenannte „unüberwachte Lernmethoden“ genutzt. Hierfür existieren keine allgemeingültigen Verfahren, die ein optimales Ergebnis garantieren.[3]Mit einer „Trial-and-Error-Methode “ können die optimalen Parameter angenähert werden.[4]. Dafür müssen Ergebnisse überprüft, Eingangsparameter angepasst sowie Entscheidung anhand von Zwischenergebnissen getroffen werden. Hierfür werden mathematische Hilfsmittel bereitgestellt, die den Nutzer bei diesen Entscheidungen unterstützen. Nachdem in Kapitel 1 die Abschlussarbeit motiviert und in Kontext gesetzt wurde, gibt Kapitel 2 einen Überblick über Energieinfrastrukturen in Deutschland. Außerdem wird in die Theorie der Clusteranalyse eingeleitet, vergleichbare Forschungsbeiträge im energiewirtschaftlichen Kontext betrachtet und die eigene Forschungsfrage bekräftigt. Kapitel 3 widmet sich detailliert den implementierten Methoden und einer dezidierten Begründung ihrer Wahl. Die Modellerstellung wird in Kapitel 4 beschrieben und in Kapitel 5 exemplarisch auf einen Datensatz des Projekts „AIRE – Anforderungen an die Infrastrukturen im Rahmen der Energiewende “ angewendet. Die einzelnen Schritte werden beschrieben, die Ergebnisse diskutiert und die Grenzen des Modell aufgezeigt. Abschließend wird in Kapitel 6 ein Fazit aus der Anwendung gezogen und ein Ausblick auf sinnvolle Erweiterungen gegeben.

[3]Caliński und Harabasz, 1974, 7 f.
[4]Davies und Bouldin, 1979, S. 227.

2 Theorie

Dieses Kapitel gibt in Abschnitt 2.1 einen Überblick über die Energieinfrastrukturen in Deutschland und deren Beeinflussung durch die Energiewende. In Abschnitt 2.2 wird kurz auf die räumliche Bezugseinheit eingegangen, auf der die Daten erfasst wurden, die in Kapitel 5 genutzt werden. Einen Überblick über die Theorie des Clusterings wird in Abschnitt 2.3 gegeben und abschließend, in Abschnitt 2.4, ausgewählte Forschungsbeiträger zur Identifikation von Typregionen im Bereich der Energiewirtschaft diskutiert.

2.1 Energieinfrastrukturen

Energie, in ihren unterschiedlichen Formen, ist ein wesentlicher Bestandteil des Alltags. Rund 70 % der Primärenergieträger unseres Energieverbrauchs werden importiert.[1] Abgesehen von Braunkohle, sind aufgrund der geographischen Lage von Deutschland und der Geologie, keine nennenswerte heimischen Vorkommen vorhanden. Fossile Primärenergieträger (Kohle, Mineralöl, Gas, etc.) werden in Kraftwerken und Raffinerien zu Sekundärenergieträgern (Strom, Gas, Treibstoff, etc.) umgewandelt und in Form von leitungsgebundenen oder festen Energieträgern an den Endnutzer geliefert. Letzteres benötigt keine eigenständige Infrastruktur und wird somit über das Verkehrsnetz verteilt. Um jeden Bürger einen direkten Zugang zu den leitungsgebundenen Energieträgern zu ermöglichen sind spezielle Transport- und Verteilinfrastrukturen notwendig. Der Energieverbrauch teilt sich in drei Sektoren auf: Strom, Wärme[2] und Verkehr. Wie in Abb. 2.1 signifikant erkennbar ist, sind diese die emissionsstärksten Sektoren. Zugleich aber auch diejenigen, die durch technologische Entwicklungen das größte Einsparpotential versprechen. Ein direkter Einfluss auf Ausbau, Umbau ggf. auch Umwidmung der Gas-, Strom und Fernwärmenetze ist gegeben. Im Folgenden wird auf diese Sektoren und ihre Infrastrukturen differenzierter eingegangen.

[1] BGR, 2018.

[2] In der Abbildung steht der Energiebedarf der Industrie hauptsächlich für industrielle Prozesswärme und wird im Folgenden zusammen mit der Gebäudewärme als Sektor Wärme betrachtet

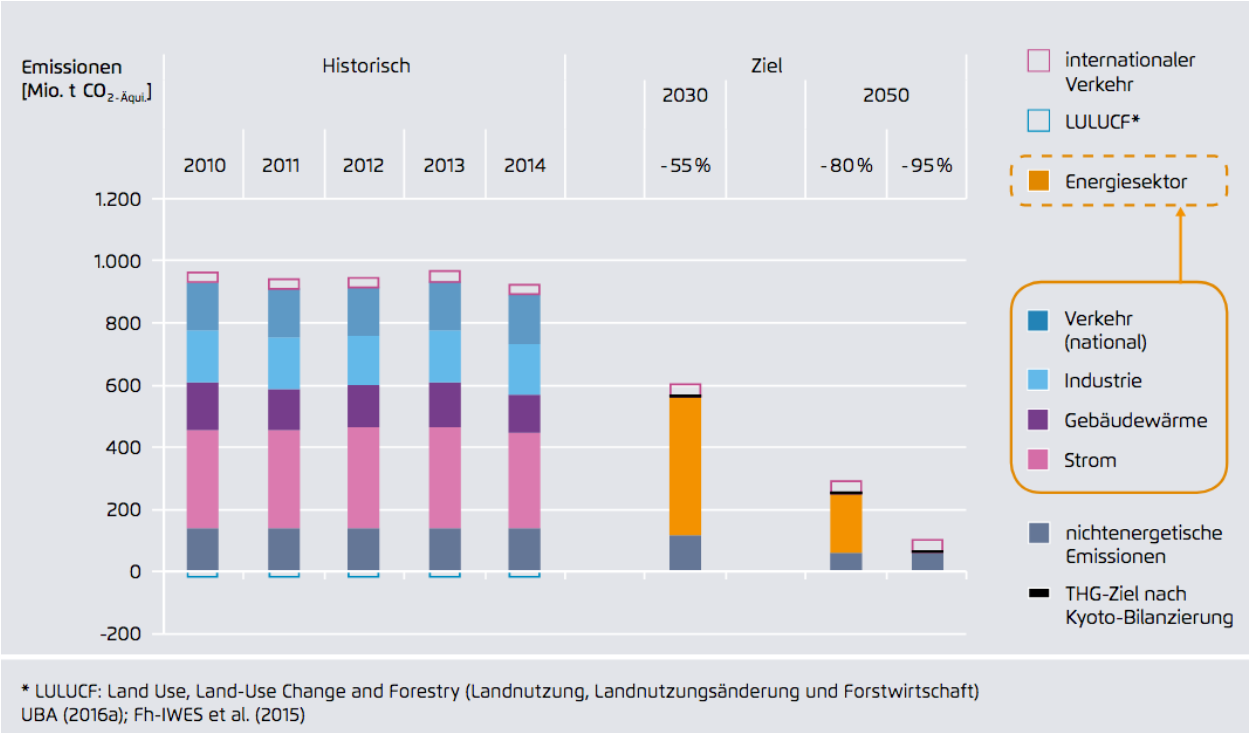


Abbildung 2.1: Emissionsbudget in Deutschland

Quelle: (IWES und IBP, 2017)

2.1.1 Strom

Über 20 % des deutschen Endenergieverbrauch ist Strom.[3] Dieser ist schwer bzw. nur in geringen Mengen speicherbar. Um das Stromnetz stabil zu halten ist deshalb ein direkter Verbrauch notwendig. Denn wenn Nachfrage und Produktion nicht ausgeglichen sind, bricht das Netz zusammen. Dies würde zu Schäden an den Verbrauchern aber auch am Netz selber führen. Aus Effizienzgründen in der Umwandlung sowie volkswirtschaftlich geringeren Kosten hat sich historisch ein zentralisiertes Stromversorgungsnetz entwickelt. Die Stromerzeugung fand in großen Kraftwerken in der Nähe von verbrauchsstarken bzw. rohstoffreichen Regionen statt um die Entfernung zum Verbraucher oder zum Rohstoff zu minimieren. In der Energiewende werden fossile Großanlagen künftig durch viele kleine dezentralen EE-Anlagen ersetzt. Abgesehen von den Off-shore Windparks speisen diese hauptsächlich im Verteilnetz ein, wie Abb. 2.2 veranschaulicht. Die verpflichtende Stromabnahme aus erneuerbaren Quellen durch die Netzbetreiber ist seit dem Stromeinspeisegesetz von 1991 gesetzlich geregelt. Dies wurde 2000 durch das Erneuerbaren-Energien-Gesetz (EEG) ersetzt und gewährte Strom aus erneuerbaren Quellen Einspeisevorrang. Die Mengenvergütung ($\frac{€}{kWh}$) führte dazu, das sich der

[3] AEE, 2019.

Ausbau der Erneuerbaren auf Regionen mit hohem Angebot fokussierte. Regionen mit hohen Vollaststunden oder hoher Solareinstrahlung versprechen einen hohen Ertrag und Gewinne. Seit 2017 ist im EEG der Zubau von EE-Anlagen einer gewissen Größe durch Ausschreibungen mit Mengendeckel geregelt. Um die Umlagen mit marktorientierten Verfahren zu verringern wurde von einer verlässlichen Einspeisevergütungen auf das Ausschreibungsverfahren umgestellt. Hierduch hat sich die Produktion und Nachfrage weiter regional entkoppelt. Der weitgehend stetigen Windproduktion im Norden steht ein stetig starker Verbrauch im Süden gegenüber. Die hohen Solarkapazitäten im Süden können nur zu Sonnenstunden einen Beitrag leisten. Als Folge des 2011 wieder beschlossenen Atomausstiegs fallen in der Nähe der Verbraucherzentren notwendige Kapazitäten weg. Um das Nord-Süd-Gefälle auszugleichen und den Transport von Strom aus Erneuerbaren Energien in diesen Zentren zu gewährleisten, werden riesige Investitionen in zusätzlichen Stromtrassen notwendig.[4]

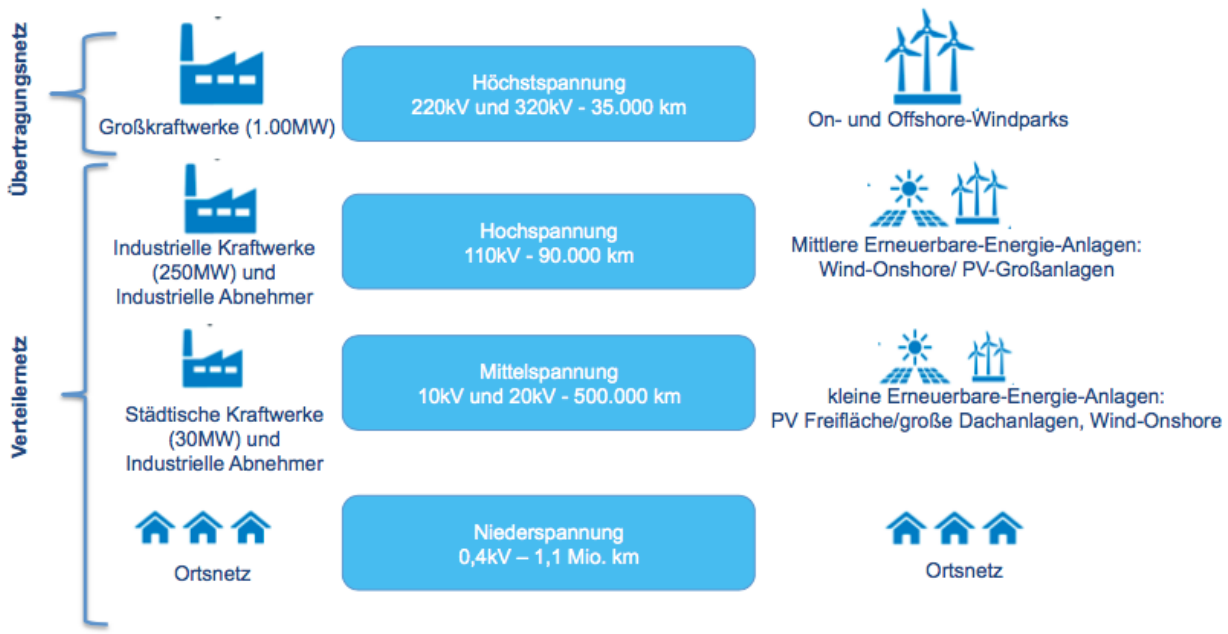


Abbildung 2.2: Verbraucher und Erzeuger im deutschen Stromnetz

Quelle: nach (BMWi, 2012)

[4] 4ÜNB, 2019.

2.1.2 Verkehr

Die primären Endenergieträger im Verkehrsbereich sind Mineralölprodukte wie Diesel und Benzin die in Raffinerien gewonnen werden.[5] Pipelines verbinden diese mit den Importhäfen (Wilhelmshaven, Brunsbüttel, Hamburg und Rostock). Mehr als die Hälfte aller Rohölerzeugnisse in Deutschland wurden im Jahr 2016 exportiert.[6] Der Export und die Verteilung zu den Tankstellen findet über Straßentankfahrzeuge, Eisenbahnkesselwagen und Tankschiffe statt. Hierbei werden bestehende Verkehrsinfrastrukturen wie Flüsse, Gleise, Straßen mitgenutzt. In den kommenden Jahren soll der Verkehr zunehmend dekarbonisiert werden. Teile des Güterverkehrs werden somit auf das elektrifizierte Schienennetz verlagert. Der Güterverkehr zur Straße wird langfristig auf Brennstoffzellentechnologie (FCEV, aufgrund der hohen Reichweite) teilweise aber auch auf batterieelektrische Fahrzeuge (BEV, bei Kurzstreckenverkehr in urbanen Regionen) umgestellt. Ebenso wird der Personenverkehr elektrifiziert (BEV, FCEV). Nach Verkehrsprognosen wird die Anzahl der Autofahrer nicht wesentlich abnehmen. Die Folge ist also, dass die entsprechenden Lade- und Tankinfrastrukturen ausgebaut werden müssen.[7]

2.1.3 Wärme

Die Wärmebereitstellung macht 53,4% des Endenergieverbrauchs in 2017 aus. Tabelle 2.1 zeigt den Anteil der Wärmeanwendungen am gesamten Endenergieverbrauch. Wärme wird beim Umwandlungsprozess eines Energieträgers übertragen und meist im Medium Wasser oder Luft transportiert. Im Falle von Gas-, Öl- oder Kohleheizungen findet die Umwandlung häufig beim Endverbraucher selbst statt.

Tabelle 2.1: Endenergieverbrauch im Sektor Wärme

Wärmeanwendungen				
	Raumwärme	**Warmwasser**	**Prozesswärme**	**gesamt**
%	26,5	4,8	22,1	100 %
Petajoule	2439,7	442,5	2037,3	9207,8 Pj

Quelle: (AGEB, 2017)

[5] UBA, 2017.
[6] BMWi, 2019a.
[7] BMVi, 2019.

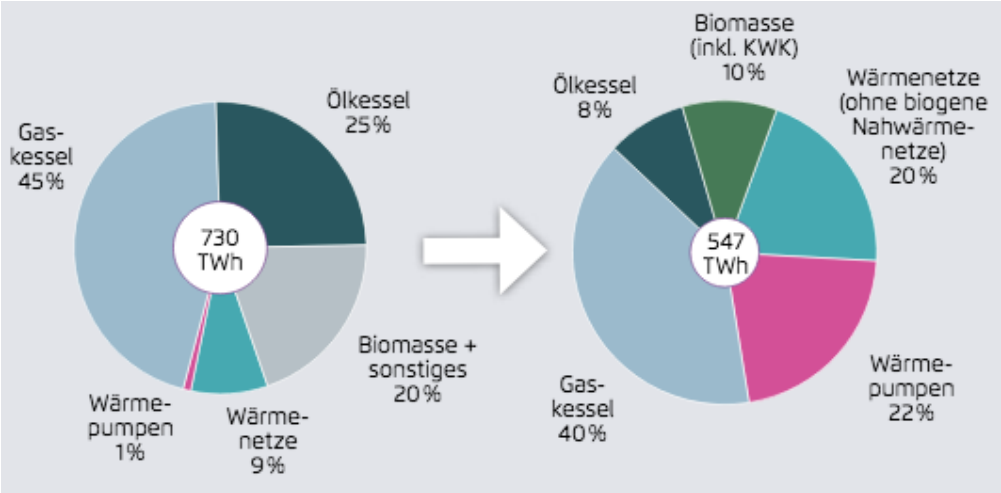


Abbildung 2.3: Gebäudewärmemix in 2015 (links) und Szenario in 2030 (rechts)

Quelle: (IWES und IBP, 2017)

Ölheizungen sind im Gegensatz zu den Energieträgern Gas, Fernwärme und Strom nicht leitungsgebunden. Große Tanks beinhalten den Heizbedarf für ein ganzes Jahr. Die Lieferung erfolgt mit dem Tanklaster über die Verkehrsinfrastruktur. Gerade in ländlichen Gebieten ohne Gasversorgung sind sie üblich. Aufgrund der politisch geforderten Emissionsreduktion im Gebäudebereich ist ein Wechsel von Öl zu Gas, Fernwärme oder Strom (Wärmepumpen) wie in Abb. 2.3 abzusehen.

Obwohl Deutschland flächendeckend mit Strom versorgt ist, war die elektrobetriebene Wärmerzeugung bis kürzlich noch keine preisgünstige Alternative. Wärmepumpen sind erst seit Kurzem in den politischen Fokus geraten. Im Rahmen der Wärmewende werden diese wegen des emissionsfreien Betriebs deutlich an Bedeutung gewinnen. Durch staatliche Förderprogramme und sinkende Investitionskosten werden sie außerdem immer attraktiver für den Endnutzer.

Auch die Fern- oder Nahwärmenutzung ist an eine Leitung zum Netz gebunden. Einige deutschen Städte haben ein Fernwärmenetz, das meist mit fossile Heizkraftwerke betrieben wird. 83 % der Endenergie wird hierbei durch Kraft-Wärme-Kopplungs-Anlagen (KWK-Anlagen) gewonnen.[8] In Neubaudörfern mit Nahwärme wird diese meist durch ein kleines Blockheizkraftwerk (BHKW) bereitgestellt. Endverbraucher beziehen die Wärme über einen Wärmetauscher von heißem Wasserdampf aus dem Netz. Überregionale Verbundnetze wie

[8]BDEW, 2015.

bei Strom und Gas existieren nicht. Die relativ günstige Wärmeerzeugung steht dem recht teuren Transportnetz gegenüber.

Tabelle 2.2: Länge der einzelnen Druckebenen das Gasversorgungsnetzes 2017

	Gasversorgungsnetz		
Drucklevel	Niederdruck	Mitteldruck	Hochdruck
Länge	157.816 km	202.983 km	121.611 km

Quelle: (BMWi, 2019b)

Gas ist aufgrund des geringen Preises der dominierende Energieträger in gasversorgten Gebieten. Ähnlich wie beim Stromnetz ist es auf mehrere Druckebenen wie in Tabelle 2.2 aufgeteilt. Da das Gasnetz schon immer auf Importe (Russland, Niederlanden, Norwegen) angewiesen und lange oligopolistisch strukturiert ist, war es bis jetzt weit weniger dem Wandel unterworfen als das Stromnetz. Deutschland besitzt ein gut ausgebautes Transportnetz zur Verbindung der regionalen Versorgungsnetze, aber auch zum Durchleiten an die europäischen Nachbarn. Der größte Erdgasverbrauch findet in den Haushalten für Raumwärme und Warmwasser (29%) sowie in der Industrie für Prozesswärme (39%) statt.[9] Fast die Hälfte der Deutschen beziehen Ihre Wärme durch Gasheizungen. Da Gas relativ emissionsarm ist, wird dies auch vorerst nicht abnehmen.

Die Industrie verursacht rund 21 % der heutigen CO2-Emissionen.[10] Diese entstehen im wesentlichen bei der Erzeugung von Prozesswärme. Hauptenergieträger sind hier Kohle (23,5 %) und Erdgas (48,9 %).(BMWi, 2019b) Nach Prognosen wird der Industriewärmebedarf auch in Zukunft auf dem heutigen Niveau bleiben.[11] Um die Klimaziele zu erreichen, soll vor alledem der Energieträger Kohle durch Wasserstoff aus Erneuerbaren Energien ersetzt werden. Dieser kann beigemischt bzw. Teile des bestehenden Netzes auf reinen Wasserstoff umgerüstet werden. Um dies zu ermöglichen besteht jedoch ein erhöhter Investitionsbedarf in die Infrastruktur.[12]

Der Wandel zu geringeren Emissionen führt zu einer zunehmenden bzw. veränderten Belastung der bereits bestehenden Infrastrukturen, die ihre etzigen Kapazitäten überschreiten(Bruns u. a., 2012). Die bestehenden Verteilnetze für Gas und Strom benötigen deshalb

[9]BMWi, 2019b.
[10]Robinius u. a., 2019, 34 f.
[11]Robinius u. a., 2019, 34 f.
[12]Bruns u. a., 2012, 203 ff.

Nach-, Um-, und Aufrüstung. Damit sich die Märkte in den Bereichen Wärme, Strom und Verkehr weiter frei und technologieoffen entwickeln können, dürfen diese nicht durch unzureichend ausgebaute Versorgungsnetze gehemmt werden. Deshalb ist es notwendig, dass der Ausbau der Netze ihren Benutzern vorausgeht. Die Herausforderung besteht hierbei in den unterschiedlichen Ausgangssituationen der einzelnen Regionen. Entwicklungsprognosen sind nur schwer abzuschätzen aber wesentliche Grundlage für wirtschaftliche Investitionen dieser Größenordnung.

2.2 NUTS-3 Klassifikation

Um Prognosen mit möglichst hoher und weiter regionaler Auflösung tätigen zu können sind einheitliche Daten auf den entsprechenden Ebene notwendig. Um europaweit einen einheitlichen Standard zur Erstellung und Verwendung von regionalen Statistiken sicherzustellen, wurden vom Statistischen Amt der Europäischen Union (Eurostat) 1981 die NUTS (The Nomenclature of Territorial Units for Statistics) Klassifikation eingeführt. Dies ermöglicht im Bereich der öffentlichen Daten, weitreichende Analysen durch Gegenüberstellung und Vergleichbarkeit von standardisierten Regionaldaten. Die Einteilung richtet sich ungefähr nach den in Tabelle 2.3 angegeben Richtwerten.

Tabelle 2.3: Richtwerte der Population der NUTS-Klassifikation

Ebene	**Untergrenze**	**Obergrenze**
NUTS 1	3.000.000	7.000.000
NUTS 2	800.000	3.000.000
NUTS 3	150.000	800.000

Quelle: (eurostat, 2018a)

Im Wesentlichen spiegelt die NUTS-Klassifikation die territoriale Verwaltungsgliederung der Mitgliedsstaaten. Die aktuelle NUTS-2016 Klassifikation, die seit 1. Januar 2018 gültig ist, umfasst:

- 104 Regionen auf NUTS-1-Ebene
- 281 Regionen auf NUTS-2-Ebene
- 1348 Regionen auf NUTS-3-Ebene

Die NUTS-1-Ebene enstpricht den 16 deutschen Bundesländern, NUTS-2 den Regierungsbezirken bzw. Stadtstaaten und Flächenländern und NUTS-3 den 401 Landkreisen bzw. kreisfreien Städten. Letztere wurden auch exemplarisch in Kapitel 5 genutzt. Datensätze bei deutschen Bundes- bzw. Landesstatistikstellen haben oft nur Regionalschlüssel, können jedoch über öffentliche Postleitzahl-NUTS Zuteilschlüssel zugewiesen werden.[13] Bei dem Transfer von NUTS-3 Schlüsseln zu älteren Daten ist eine erhöhte Achtsamkeit bezüglich der Kreisreformen gegeben, da die Klassifizierung nur im 3-Jahres-Rhythmus angepasst wird. Hervorzuheben ist hier die letzte Anpassung von 2018, die auf eine Gebietsreform von 2016 zurückzuführen ist.

NUTS-2013 zu NUTS-2016 Die NUTS-Regionen DE915 (Göttingen) und DE919 (Osterode am Harz) wurden zusammengelegt zu DE91C (Göttingen)(AGS: 03159). Außerdem wurde der NUTS-3 Code der Region DEB16 (Cochem-Zell, Gemeindeschlüssel 07135) zu DEB1C und der Code der Region DEB19 (Rhein-Hunsrück-Kreis, Gemeindeschlüssel: 07140) zu DEB1D umcodiert.

Standardisierte und öffentliche Daten sind Grundlage zahlreicher Studien und Forschungsprojekte, die den Transformationsprozess begleiten. Jedoch sind viele Erkenntnisse durch den begrenzten lokalen Untersuchungsraum und der Diversität von Regionen nicht aufeinander übertragbar. Um allgemeinere Aussagen treffen zu können, muss der Untersuchungsraum vergrößert werden. Aufgrund der Vielzahl von Einflussgrößen scheint es sinnvoll mit Komplexitätsreduktion zu arbeiten.

2.3 Clusteranalyse

Bei der Betrachtung einer Objektmenge, können sich ähnelnde Objekte, in homogene Klassen zusammengefasst werden. Das ermöglicht eine Komplexitätsreduktion der gesamten Struktur. Dieses Vorgehen wird in der Abschlussarbeit verwendet. Die Entwicklung einer Methodik zur Identifikation von Typregionen mit dem Clusteralgorithmus K-Means ist dabei der Hauptbestandteil. Im Folgenden wir ein Überblick über die Clusterverfahren gegeben, die es ermöglichen, komplexe Sachverhalte strukturell zu vereinfache. Zunächst werden jedoch in

[13] eurostat, 2018b.

Tabelle 2.4 Begriffe definiert, die für den weiteren Zusammenhang elementar sind. Kapitel 3 wird dann noch einmal auf die spezifisch angewendeten Methoden eingegangen.

Tabelle 2.4: Begriffsdefinition in der Clusteranalyse

Begriff	**Beschreibung**	**Tabellenbezug**
Sample	Beobachtung/Objekt/Ding	Zeilen
Feature	Variable oder Eigenschaft der Samples	Spalten
Wert	numerischer Wert eines Features für ein Sample	Zelle
Set	Menge an Features gleicher Anzahl an Samples	Tabelle
Cluster	Klasse oder Gruppe, Menge von Samples	Gruppe von Zeilen

2.3.1 Distanzmaße

Clusteralgorithmen arbeiten mit Hilfe von mathematischen Algorithmen um Strukturen oder Gruppen in Daten aufzuspüren. Diese Verfahren wurden in den 1970er Jahren entwickelt. Forschende nutzten hier bereits die ersten Computer als Hilfsmittel zum Lösen dieser Algorithmen. Bei kategorialen Features werden Ähnlichkeitsmaße verwendet. Bei metrischen Features definieren sich die Gruppen durch Nähe zu oder Distanz voneinander. Da in dieser Arbeit ausschließlich mit metrischen Daten gearbeitet wurde, ist die Definition eines einheitlichen Abstandsbegriffs (Metrik) notwendig.[14]

Metrik: Eine Relation $dist()$ zwischen zwei Punkten i und j heißt Metrik auf dem kartesischen Produkt $S \times S$ wenn nach Wiedenbeck und Züll folgende Axiome erfüllt sind:[15]

1. Positive Definitheit: $dist(i,i) = 0 \qquad \forall\, i,j \in S$
2. Symmetrie: $dist(i,j) = dist(j,i) \geq 0 \qquad \forall\, i,j \in S$
3. Dreiecksungleichung: $dist(i,j) \leq dist(i,k) + dist(k,j) \qquad \forall\, i,j \in S$

(1) Ist der Abstand null, sind die Punkte identisch. (2) Der Abstand muss von beiden Punkten ausgehend gleich und größer gleich null sein. (3) Außerdem beschreibt sie den direkten Weg und dieser ist immer der kürzeste. Mit Hilfe dieser Metriken kann der Abstand zwischen zwei beliebigen Punkten in einem belieben Raum bestimmt werden. Um zu vermeiden, dass die Werte beim Vergleichen durch willkürlich gewählte Maßeinheiten beeinflusst werden, sollte

[14] Hastie u. a., 2009.
[15] Wiedenbeck und Züll, 2010, 535 f.

auf die Unveränderlichkeit bei unterschiedlichen Skalen geachtet werden. Wichtige Invarianz-Forderungen an diese Maße sind:

Translationsinvarianz: Sie besagt, dass die Distanz nicht von der Wahl des Koordinatenursprungs beeinflusst ist. Der Abstand zweier Punkte verändert sich also nicht, wenn diese einer Translation T um einen bestimmten Verschiebungsvektor $\overrightarrow{v}$ unterzogen werden.

$$\begin{aligned} f &: \mathbb{R} \rightarrow \mathbb{R} \\ T\, f(x) &= f(x-a) \end{aligned} \tag{2.1}$$

Skaleninvarianz: Skaleninvariante Distanzmaße werden nicht von einer Skalierung beeinflusst. Trotz der Änderung der Betrachtungsgröße verändert sich die Distanz somit nicht.

$$\begin{aligned} f &: \mathbb{R} \rightarrow \mathbb{R} \\ f(ax) &= C(A) f(x) \end{aligned} \tag{2.2}$$

Bei der Wahl des Distanzmaßes ist es wesentlich, dass die Invarianzeigenschaften mit der Semantik der Daten übereinstimmen. Von der Bedeutung der Daten sollte auf die Anforderungen geschlossen werden. Viel genutzte Metriken leiten sich von der Minkowski-Metrik in Gleichung (2.3) ab. Diese sind von den p-Norm Metriken aus dem 2-dimensionalen Raum für Reele Zahlen $\mathbb{R}^2$ abgeleitet.[16]

$$dist_p(i,j) = \sqrt[p]{\sum_{i=1}^{n} \mid x_i - x_j \mid^p} \tag{2.3}$$

hier steht q für die Dimensionszahl des Raumes und p für den Metrikparameter. Für den Minkowski-Parameter p=2 ergibt sich die euklidische Distanz.[17] Diese entspricht unserer

[16] Stein und Vollnahls, 2012.
[17] Stein und Vollnahls, 2012.

übliche geometrischen Abstandvorstellung und wird häufig bei intervall- oder verhältnisskallierten Werten genutzt. Jedoch erfüllt sie nicht das Kriterium der Skaleninvarianz. Deshalb müssen die Daten vor ihrem Vergleich normiert werden (Abschnitt 3.3).

$$dist_2(i,j) \;=\; \sqrt[2]{\sum_{i=1}^{p}(x_{ik}-x_{jk})^2} \;=\| \; x_i \; - \; x_j \; \| \tag{2.4}$$

Da die euklidische Distanz auch invariat gegenüber orthogonalen Transformationen ist, kann sie in Kombination mit dimensionsreduzierenden Verfahren wie der Hauptkomponentenanalyse verwendet werden.

2.3.2 Fluch der Dimensionen

Beim sogenannten „Fluch der Dimensionen“[18] wird durch eine zu hohe Anzahl an Dimensionen (Features) bei verhältnismäßig kleiner Anzahl an Beobachtungen (Samples), die Genauigkeit des Ergebnisses stark beeinflusst. Durch zusätzliche Dimensionen steigt das Volumen rapide an und es werden deutlich mehr Samples benötigt um klare Strukturen zu schaffen. Die kleinste und größte Distanz unterscheiden sich kaum mehr voneinander. Es entsteht ein Rauschen in den Daten, anspruchsvollere Maßnahmen notwendig macht. Laut Zimek u. a.[19] muss ein schlechtes Verhältnis von Features/Samples nicht zwangsweise zu schlechten Ergebnissen führen. Lediglich zusätzliche Features mit redundanter Information haben einen negativen Einfluss. Durch ein sorgfältiges Entfernen von unnötigen Features und Ausreißern kann dem entgegen gewirkt werden.

2.3.3 Clusterverfahren

Clusterverfahren werden genutzt, um den Daten zugrundeliegende Strukturen, zu entdecken. Dabei wird in einem iterativen Prozess die Anzahl der Cluster, die Größe oder die Zusammensetzung verändert bis ein Abbruchkriterium erfüllt ist. Im Wesentlichen unterscheiden sich traditionell zwei Verfahrensarten beim Clustern:

[18] Bellman, 2015, S. 94.
[19] Zimek u. a., 2012.

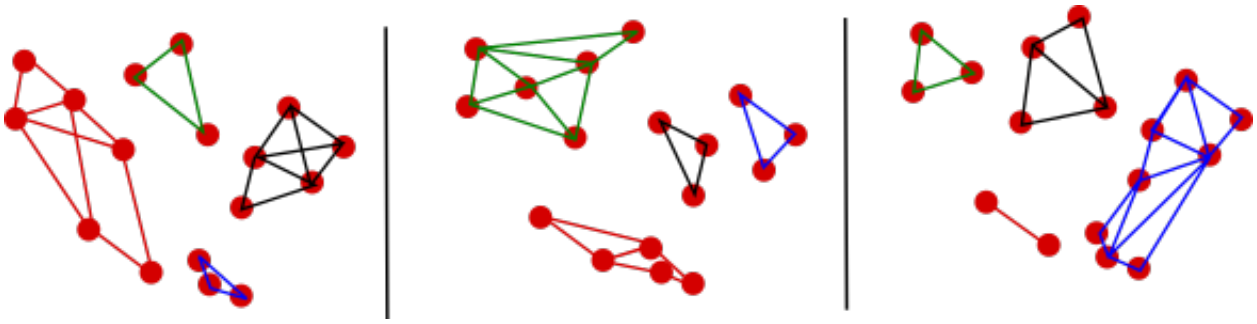


Abbildung 2.4: Schematische Darstellung unterschiedlicher Clusterzugehörigkeiten in partitionierenden Verfahren bei k=4

Quelle: eigene Darstellung

partitionierende Verfahren Diese Verfahren zeichnen sich dadurch aus, dass sie den Datensatz in einen vorgegebene Anzahl an Cluster partitionieren. Hierfür werden Clusterzentren bestimmt und jedes Sample dem nächsten Zentrum zugeordnet. Anschließend wird aus den Values der Cluster-Samples ein neues Zentrum bestimmt und die Samples erneut zugeordnet. Das Zentrum wird dadurch so lange verschoben bis sich die Zuordnung nicht mehr ändert oder die Abweichung gegen einen vordefinierten Wert konvergiert. Wie in Abb. 2.4 können die Samples somit ihre Clusterzugehörigkeit ändern.[20] Einzelne Variationen unterscheiden sich in der Bestimmungsart der Clusterzentren oder ihrer Zuordnungsart. Weiche Zuordnungen zeichnen sich durch einen Zugehörigkeitswert aus. Dieser liegt zwischen [0,1] und existiert für jedes Samples zu jedem Cluster. Harten Zuordnung hingegen sind eindeutig und binär. Die wesentlichen Parameter die Einfluss haben, sind somit die Anzahl k der Cluster, die Formulierung zur Zentrumsberechnung sowie das Abbruchkriterium.[21] Ein gängiger Algorithmus ist K-Means, welcher das neue Clusterzentrum aus den Mittelwerten bestimmt. Auf diesen wird in Abschnitt 3.4 noch genauer eingegangen. Andere Varianten sind K-Median, das den Medianwert als Zentrum bestimmt. K-Mediod hingegen, legt das Zentrum in ein existierendes Sample, das dem Mittelwert am nächsten ist.

hierarchische Verfahren Hierarchischen Clusterverfahren unterteilen sich in agglomerative und divisive Verfahren. Agglomerative Verfahren fassen immer mehr Cluster zusammen. Begonnen wird bei der höchsten Anzahl von Clustern (Anzahl der Features). Iteration für Iteration werden dann, wie in Abb. 2.5, zwei weitere Cluster bzw. Objekte zu einem Neuen

[20] Wiedenbeck und Züll, 2010, 527 ff.
[21] Jäckle, 2017.

fusioniert. Dabei nimmt die Anzahl der Cluster ab aber die Anzahl der Samples pro Cluster zu. Beim divisiven Verfahren funktioniert dies umgekehrt und Cluster werden immer weiter aufgeteilt. Die Homogenität der Cluster nimmt somit stetig zu bzw. ab.[22] Der Fokus liegt also nicht auf dem Abstand einzelner Samples zum Zentrum sondern auf dem Abstand der Cluster zueinander. Hierbei muss keine feste Anzahl von Cluster vorgegeben werden. Entscheidend ist die Wahl des Fusionierungsalgorithmus und der Schwellenwert bei dem fusioniert werden soll. Beliebt ist das 'Single-Linkage' Verfahren. Hierbei werden die Cluster der zwei nächsten Samples verbunden. Dieses Verfahren eignet sich auch zum Identifizieren von Ausreißern, da diese als letztes fusioniert werden. Ein weiteres Verfahren ist im Ward-Algorithmus umgesetzt. Hierbei wird um das Sample erweitert wird, was die geringste Änderung in der Varianz innerhalb des Clusters zur Folge hat.

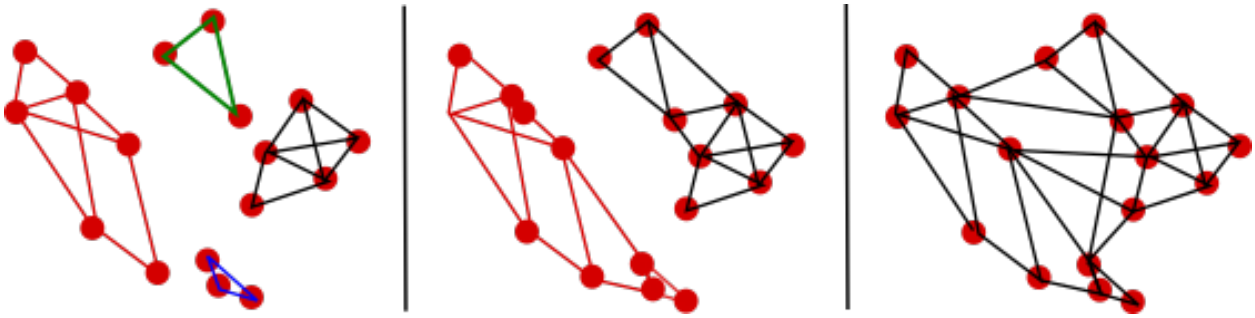


Abbildung 2.5: Schematische Darstellung eines agglomerativen Verfahrens

Quelle: eigene Darstellung

2.3.4 weitere Verfahren

In den 1990er Jahren wurden noch weitere Verfahren entwickelt. Dichtebasierte Verfahren beziehen sich auf die Anordnung der Samples in mehrdimensionalen Räumen. Unterschieden wird zwischen Räumen die dicht gruppiert sind und Räumen mit einer geringen Anzahl an Samples, die dichte Räume voneinander trennen. Beim DBSCAN-Algorithmus muss jedes Samples, dass zu einem Cluster gehören will, eine Mindestanzahl an Samples im radialen Umfeld haben. Das nachträgliche Hinzufügen und Entfernen von Samples ist einfach umzusetzen, da es nur Auswirkung auf seine unmittelbare Nachbarschaft hat. Durch diese Herangehensweise haben auch Ausreißer kaum Auswirkung auf die Clusterbildung. Es muss jedoch in Kauf genommen werden, dass einige Samples keinem Cluster zugeordnet werden. Außerdem ist die Anzahl dieser Sample sehr stark vom Tuning der Parameter abhängig.

[22] Wiedenbeck und Züll, 2010, 527 ff.

Auch aus diesem Grund werden sie gerne in großen Multimediadatenbänken genutzt wo man die Parameter über lange Zeit anpassen kann.[23] Gitterbasierte Verfahren wurden für Geodaten entwickelt. Diese rastern die mehrdimensionalen Räume, berechnen für die einzelnen Rasterzellen statistische Parameter und clustern anschließend diese Informationen mit einem hierarchischen Verfahren. Obwohl es für Geodaten entwickelt wurde findet es auch Anwendung in der Frequenzwellentechnik und Weiteren.[24]

Des weiteren gibt es die Möglichkeit unterschiedliche Verfahren miteinander zu kombinieren. Hierarchische Verfahren bieten sich zum identifizieren von Ausreißern an und können beispielsweise vor ein partitionierendes Verfahren geschaltet werden. Bei großen Datensätzen wird oft mit einem Trainingsdatensatz gearbeitet, der ca 80 % aller Samples ausmacht. Der Vorteil besteht in einem geringeren Overfitting, also einem geringern Einfluss von Ausreißern. Bei kleinen Datensätzen muss oft darauf verzichtet werden. Alternativ werden Ausreißer mit gesonderten Verfahren identifiziert und entfernt.

2.4 Forschungsstand

Heutzutage erscheint das Gruppieren von Informationen bei der Unmenge an Daten, die überall erfasst werden, sinnvoll. Obwohl Clusterverfahren im weit gegriffenen Energiewirtschaftsbereich häufig zur Anwendung kommen, ist die Anwendung dieser Methode beim Identifizieren von kontextbezogenen Typregionen selten genutzt. Ein möglicher Grund dafür ist die geringe Verfügbarkeit von empirisch erfassten und öffentlich zugänglichen Daten. Statistisch abgeleitete Daten können aufgrund ihrer Korrelation zu deutlich abweichenden Ergebnissen führen .[25] In seiner Dissertation analysiert (Wall, 2016) kreisfreie Städte in Deutschland anhand von Energie-Indikatoren. Er vergleicht mehrere Clusterverfahren sowie Sets an Indikatoren. Aufgrund der kleinen Objektmenge und dem somit schlechten Verhältnis von Samples (103) zu Features (51) nutzt er die Faktorenanalyse zur Dimensionsreduktion. Letztlich werden elf Faktoren standardisiert und mit dem Ward-Algorithmus geclustert. Abschließend interpretiert er die gefundenen Cluster und setzt sie in Kontext. In (Geyler u. a., 2008) wird eine etwas größere Menge von 250 Gemeinden anhand von 16 Kennzahlen der

[23] Halkidi u. a., 2001, S. 8.
[24] Halkidi u. a., 2001, S. 4.
[25] Sambandam, 2013.

Siedlungs-, Wirtschafts- und Bevölkerungsstruktur geclustert. Die Daten wurden standardisiert und ebenfalls mit dem Ward-Algorithmus bearbeitet. Der Schwerpunkt der Arbeit liegt weniger auf der Methodik als auf der Interpretation der Ergebnisse. In der Masterarbeit von (Linsenmeier, 2017) werden Netzdaten synthetischer, regionaler Verteilnetze mit einem überwachten (Regressionsbäumen) und einem unüberwachten Verfahren (K-Means) verglichen. In einem Set von 2928 Samples werden 50 optimale Cluster identifiziert. Typregionen wurden identifiziert und dann als exemplarisch für die Cluster angesehen um die Netzausbaukosten abzuschätzen. Hierbei wurden mehrere Ausbauszenarien miteinander verglichen. Der Autor schlussfolgert letztlich, dass bei der Datenlage und bei dem gegebenen Kontext, Regressions-Bäume die bessere Methode darstellen. Eine weitere, sehr aktuelle und ausführliche Arbeit von (Weinand u. a., 2019) beschäftigt sich mit der Identifikation von Gemeindetypen die Energieautarkie anstreben. Es wird sowohl auf Vergleichsliteratur, den Datensatz, die Methodik als auch die Ergebnisse ausführlich eingegangen. Aus den zuerst 59 Indikatoren, schließt er 21 Indikatoren aufgrund von Abhängigkeiten aus. Mit Hilfe einer Minimum-Maximum-Standardisierung und Faktorenanalyse reduziert er den Datensatz. Mit dem Ward-Algorithmus werden somit 11,131 Gemeinden anhand von 10 Faktoren geclustert. Um die optimale Clusteranzahl zu finden, vergleicht Weinand u. a. 30 Methoden, die abweichende Ergebnisse liefern. Er entscheidet sich für 10 Cluster für die Typregionen identifiziert werden.

Die hier genannten Forschungsergebnisse zeigen, dass die Clusteranalyse zur Identifikation von Typregionen auch im energiewirtschaftlichen Kontext Anwendung findet. Unterschiede in der Methodik sind aufgrund der individuellen Anpassung auf den Datensatz immer gegeben, jedoch lassen sich oft genutzte Verfahren identifizieren. Es konnte jedoch kein Modell gefunden werden, das die Anwendung von gängigen Cluster- und Valididerungsmethoden auf unterschiedliche Datensätze vereinfacht. Dabei erscheint es mit der steigenden Komplexität der Modelle und zunehmender Datenmenge durchaus sinnvoll. Aufgrund der breiten Anwendbarkeit und eines quasi-standardisierten Verfahren würde dies Forschungsfragen, die sich auf die Identifikation von Typregionen beziehen deutlich vereinfachen. Außerdem würde es eine Vergleichbarkeit der Anwendungen unterstützen. Dieses Model beschränkt sich vorerst auf den Algorithmus K-Means. Die Implementierung is jedoch modular und kann um zusätzliche Verfahren erweitert werden.

3 Methodik

Dieses Kapitel beschäftigt sich mit den im Modell angewendeten Methoden. Die wesentlichen Schritte des Modells werden in Abb. 3.1 illustriert:

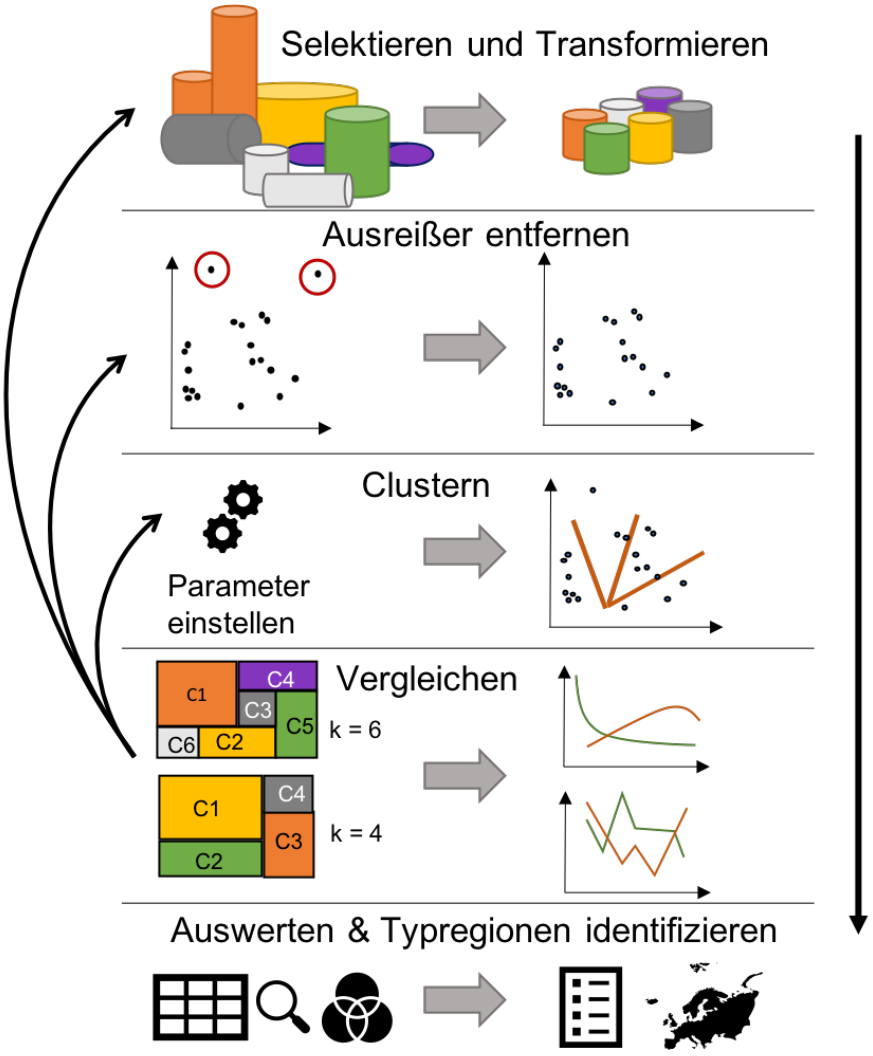


Abbildung 3.1: Wesentliche Schritte des Modellablaufs

Quelle: eigene Darstellung

Grundlage einer jeden Analyse ist die Aufbereitung des Datensatzes. Wie in Abschnitt 3.1 beschrieben, erlangen Analysierende erste Informationen über Beschaffenheit und Semantik der Daten. Zusammen mit den Werten der Korrelationsanalyse aus Abschnitt 3.2 kann dann eine sinnvolle Auswahl an Features getroffen werden. Mit den Methoden aus Abschnitt 3.3 werden die Features transformiert um ihre Skalen vergleichbar zu machen. Dies ist sowohl für

die Clusteranalyse als auch die Auswertung von Zwischen- und Endergebnissen notwendig. Abschnitt 3.4 erklärt den verwendeten Algorithmus K-Means. In Abschnitt 3.5 werden die Indizes beschrieben, die als Hilfsmittel zur Bestimmung der Clusteranzahl und Clustergüte genutzt werden. Hierbei wird die Nomenklatur aus Tabelle 2.4 weiter angewendet.

3.1 Datengrundlage

Bei Datenanalysen - und besonders bei Cluster-Analysen - ist es wichtig, dass Nutzende sich mit der Herkunft, dem Inhalt sowie deren Semantik vertraut macht. Das Clustering kann zu unterschiedlichen Ergebnissen führen, abhängig von den spezifischen Parametern die verwendet werden. Indizes (Abschnitt 3.5) sind ein hilfreiches Mittel um die Parameter richtig zu setzen. Sie können aber auch trügerisch sein, wenn sie nicht zum Datensatz passen. Das Aufbereiten und Validieren des Datensatzes stellt oft eine mühselige Arbeit dar, ermöglicht es Analysierenden aber auch die Daten kennenzulernen. Sind einzelne Werte nicht vorhanden, können diese mittels Regression oder Median- bzw. Mittelwertimputation aufgefüllt werden. Bei einer Vielzahl an fehlenden Werten sollte das Feature gegebenenfalls ausgeschlossen werden. Auf folgende Punkte ist bei der Datenaufbereitung deshalb zu achten:

- Genauigkeit/Rauschen
- Vollständigkeit
- Konsistenz/Widerspruch
- Aktualität
- Mehrwert/Redundanz

In dieser Arbeit wird mit einem bestimmten und vollständigen Datensatz aus dem Projekt „AIRE“ gearbeitet. Alle Einflussfaktoren sind vollständig vorhanden und unterscheiden sich lediglich in ihren Werten, Varianz und Verteilung.

3.2 Korrelationsanalyse

Beim Überprüfen von Abhängigkeiten fokussiert sich die Arbeit ausschließlich auf die bivariate lineare Korrelation. Dabei wird untersucht, wie ähnlich sich zwei Variablen X und Y hinsichtlich einer linearen Abhängigkeit sind. Wenn Variablen perfekt korrelieren, stellen sie in der Clusteranalyse die selbe Information doppelt da. Dadurch erhält die Information das doppelte Gewicht und verzerrt das Ergebnis.[1] Da alle Features gleich gewertet werden sollen

[1] Sambandam, 2013.

ist eine Korrelation meist unerwünscht. Besonders bei hoch dimensionalen Analysen ist der Einfluss dann schwer abzuschätzen. Um die Features im Modell auf Korrelation zu prüfen können drei verschiedene Koeffizienten genutzt werden:

- r Pearson's Korrelationskoeffizient
- ρ Spearman's Rangkorrelationskoeffizient
- τ Kendall's Konkordanzkoeffizient

Alle Koeffizienten haben den gleichen Wertebereich zwischen 1 und -1. Der Wert 0 spricht gegen eine Korrelation. Der Wert 1 bzw. -1 steht für eine perfekt bzw. umgekehrt perfekte Korrelation.

3.2.1 Pearson

Der Produkt-Moment-Korrelationskoeffizient r_{XY} nach Pearson kann nur bei intervallskalierten verwendet werden. Er ist eine besondere Form der Kovarianzmatrix[2] $cov(X, Y)$. Wie in Gleichung (3.1) beschrieben, werden die Werte standardisiert, indem sie durch die Standardabweichung σ_X und σ_Y beider Features dividiert werden.[3] Der Erwartungswert ergibt sich daher zu $E = 0$, die Varianz zu $V = 1$ und der Koeffizient ist somit skaleninvariant (2.3.1):[4]

$$r_{XY} = \frac{cov(X, Y)}{\sigma_X \cdot \sigma_Y} \tag{3.1}$$

$$r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.2}$$

mit x_i und y_i für die Werte der Features, $\bar{x}$ und $\bar{y}$ als die gemittelten Werte aller n Samples je Feature.[5] Da in Gleichung (3.2) der Mittelwert verwendet wird, reagiert der Koeffizient nicht besonders robust auf Ausreißer. Dies sollte grundsätzlich bei der Verwendung bedacht werden. Außerdem müssen bei niedrig skalierten Daten rang-korrelierende Verfahren, wie das von Kendall oder Spearman, verwendet werden. Dies gilt auch bei Daten die nicht

[2] auch Streuungsmatrix genannt
[3] Wiedenbeck und Züll, 2010, S. 86.
[4] Bamberg u. a., 2017.
[5] Backhaus u. a., 2016.

annähernd normalverteilt sind. Der Lillifors-Test[6] kann dies prüfen, wird hier aber nicht weiter behandelt.

3.2.2 Spearman

Der Spearmansche Rangkorrelationskoeffizient ρ_{XY} ist eine Abwandlung des Pearson-Koeffizienten. Hierbei werden die Daten zuerst in Ränge[7] konvertiert bzw. rang-transformiert und dann berechnet. Deshalb lässt er sich sowohl auf metrische, als auch auf ordinal-skalierte Daten anwenden.[8] Die Berechnung erfolgt durch Gleichung (3.3).

$$\rho_{XY} = \frac{\sum_{i=1}^{n}(R(x_i) - \bar{R}_x)(R(y_i) - \bar{R}_y)}{\sqrt{\sum_{i=1}^{n}(R(x_i) - \bar{R}_x)^2}\sqrt{\sum_{i=1}^{n}(R(y_i) - \bar{R}_y)^2}} \tag{3.3}$$

mit $R(x_i)$ als Rang des Samples x_i und $\bar{R}_x$ als mittlerer Rang des Features X. Die Werte dieses Koeffizienten fallen meist etwas geringer aus, als die von Pearson.

3.2.3 Kendall

Auch Kendall's Konkordanzkoeffizient τ kann bei intervall- oder ordinal-skalierten Daten angewendet werden. Er nutzt die Unterschiede in den Rängen, vergleicht und bewertet diese. Es werden immer Paare von aufeinanderfolgenden Samples betrachtet. Ist die Rangabfolge der beiden Features identisch, gehören sie zu den konkordanten Paaren C, ansonsten zu den unkonkordanten Paaren D.[9] Ist der Rang aufeinanderfolgender Samples in X oder Y identisch, besteht eine Bindung T_X bzw. T_Y. Die Anzahl dieser Paarbewertungen fließt dann in den Koeffizienten $\tau_{X,Y}$in Gleichung (3.4) ein.[10]

$$\tau_{XY} = \frac{\mid C \mid - \mid D \mid}{\sqrt{(\mid C \mid + \mid D \mid + \mid T_X \mid) \cdot (\mid C \mid + \mid D \mid + \mid T_Y \mid)}} \tag{3.4}$$

[6] Lilliefors, 1967.
[7] Der Rang ergibt sich aus der Position des Samples wenn das ganze Feature der Größe nach sortiert wird. Mehrere Samples können ranggleich sein.
[8] Bamberg u. a., 2017.
[9] Wiedenbeck und Züll, 2010, S. 84.
[10] Kendall, 1970.

3.2.4 Feature-Selection

Bei der Feature-Selection wird eine Auswahl an Features getroffen die als Untermenge weiter verwendet werden soll. Dabei wird von der Annahme ausgegangen, dass die vorhandenen Daten redundant sind und somit gewisse Features keinen Mehrwert an Information bieten bzw. sogar das Ergebnisse verzerren. Mit der Korrelationsanalyse werden Paare identifiziert. Feste Grenzwerte existieren jedoch nicht und Empfehlungen reichen je nach Quelle von 0,4 bis 0,8.[11] Eines der Feature wird entfernt. Die Auswahl ist individuell und von dem Ziel der Clusteranalyse abhängig. Die Selektion kann grundsätzlich auch aus inhaltlichen Gründen erfolgen. Per „Trial-and-Error“ wird dann ausprobiert ob der Verbleib des Features informationstragend ist bzw. inwiefern das Ergebnis beeinflusst wird.

3.3 Transformation

Beim unüberwachten Lernen hat die Skalierung der Features eine großen Einfluss auf das Ergebnis. Hat ein Feature eine Varianz, die um ein Vielfaches größer ist, kann sie die Zielfunktion des Algorithmus dominieren. Deshalb ist es von großer Bedeutung, die Eingangsdaten so zu skalieren, dass ihre Variabiliät[12] der Semantik[13] entspricht oder ihr zumindest nicht widerspricht. Um die Features in eine vergleichbare Form zu bringen, gibt es mehrere Transformationsmethoden. Diese nutzen, entsprechend ihres Kontextes, unterschiedliche Formen der Normalisierung bzw. Standardisierung. Um die Unterschiede zu verdeutlichen, die Begriffe werden kurz voneinander getrennt:

Normalisierung Beim Normalisieren wird ein Vektor[14] durch eine Norm eines Vektors dividiert, um dessen Länge auf einen bestimmten Wert zu setzen. Oft wird hier die Neuskalierung durch das Minimum und der Länge des Vektors genutzt, damit alle Elemente zwischen 0 und 1 liegen.[15]

Standardisierung Beim Standardisieren wird von einem Vektor ein Maß für die Position subtrahiert und anschließend durch ein Maß für die Größe dividiert. So wird dessen Position

[11] Herink und Petersen, 2004.
[12] der Unterschied zwischen den einzelnen Varianzen der Feature
[13] die Bedeutung der Daten hinsichtlich dessen was Sie ausdrücken oder ausdrücken sollen
[14] Feature
[15] comp.ai.neural-nets, 2014.

verändert und die Länge auf einen bestimmten Wert gesetzt. Somit ist die Standardisierung eine Verschiebung und eine Normalisierung.[16]

Anhand von 2 Features eines Beispieldatensatzes aus Sci-kit-learn soll im Folgenden, die Auswirkung der erklärten Methoden veranschaulicht werden. Die Rohdaten sind in Abb. 3.2 zu sehen. Der Fokus bei den Vergleichen liegt auf dem Einfluss der Ausreißer auf die Skalierung und somit deren Vergleichbarkeit nach der Transformation. Die Grafik besteht aus zwei Streudiagrammen für die Feature X1 und X2. Die Cluster-Zuordnung wird über die Farbgebung dargestellt. Im linken Diagramm werden alle Samples abgebildet. Da im rechten Diagramm das erste und letzte Perzentil[17] ausgeschlossen sind, ist die Achsenskalierung an die Hauptmenge der Daten angepasst. Zusätzlich ist die Verteilung, den Skalen entsprechend, an den gegenüberliegende Achsen abgebildet. Die Werte von X1 sind relativ gleichmäßig im Bereich von $[0 - 15]$ verteilt und beinhalten keine Ausreißer. X2 hat hingegen mehrere im drei- und vierstelligen Bereich obwohl sich die Hauptmenge im Wertebereich von $[0 - 5]$ befindet. Somit eignen sich diese Features ideal um den Einfluss der Ausreißer auf die Transformationsmethoden abzubilden.

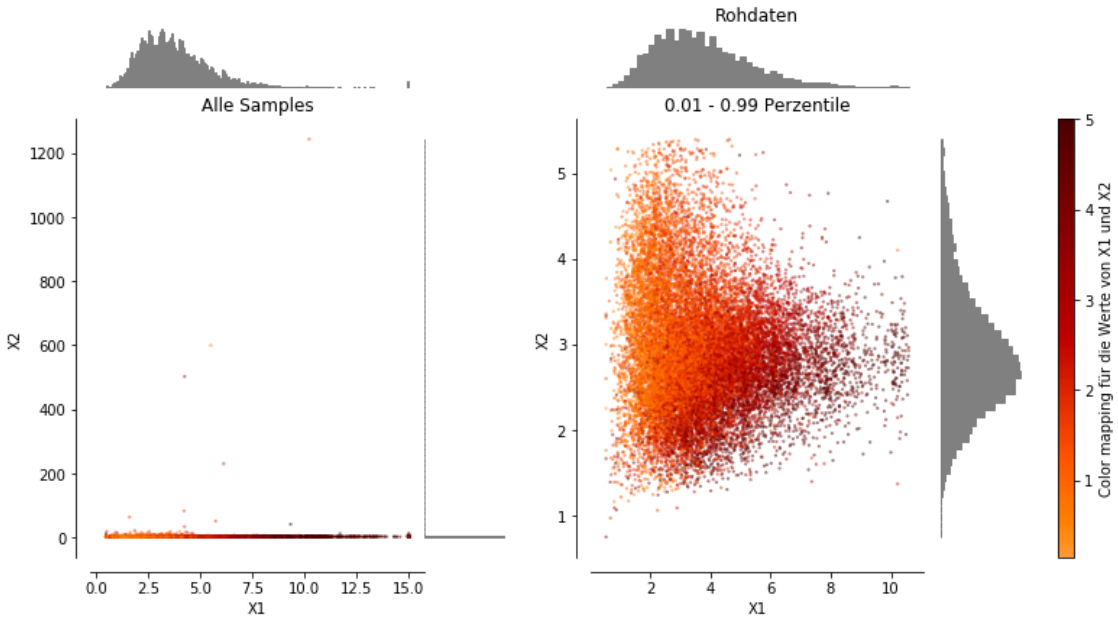


Abbildung 3.2: unbearbeiteter Datensatz

Quelle: nach (Raghav u. a., 2019)

[16] comp.ai.neural-nets, 2014.
[17] Quantile von 0,01 bis 0,99 in Schritte von 0,01

3.3.1 z-Transformation

Die z-Transformation ist eine viel genutzte Form der Standardisierung. In Gleichung (3.5) wird der Erwartungswert[18] μ subtrahiert um die Variable auf null zu zentrieren und anschließend durch die Standardabweichung σ dividiert.[19] Dadurch wird eine Zufallsvariable Z mit Erwartungswert $\mu = 0$ und Standardabweichung $\sigma = 1$ erhalten.

$$Z = \frac{X - \mu}{\sigma} \tag{3.5}$$

$$\sigma = \sqrt{s^2} \tag{3.6}$$

$$s^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{3.7}$$

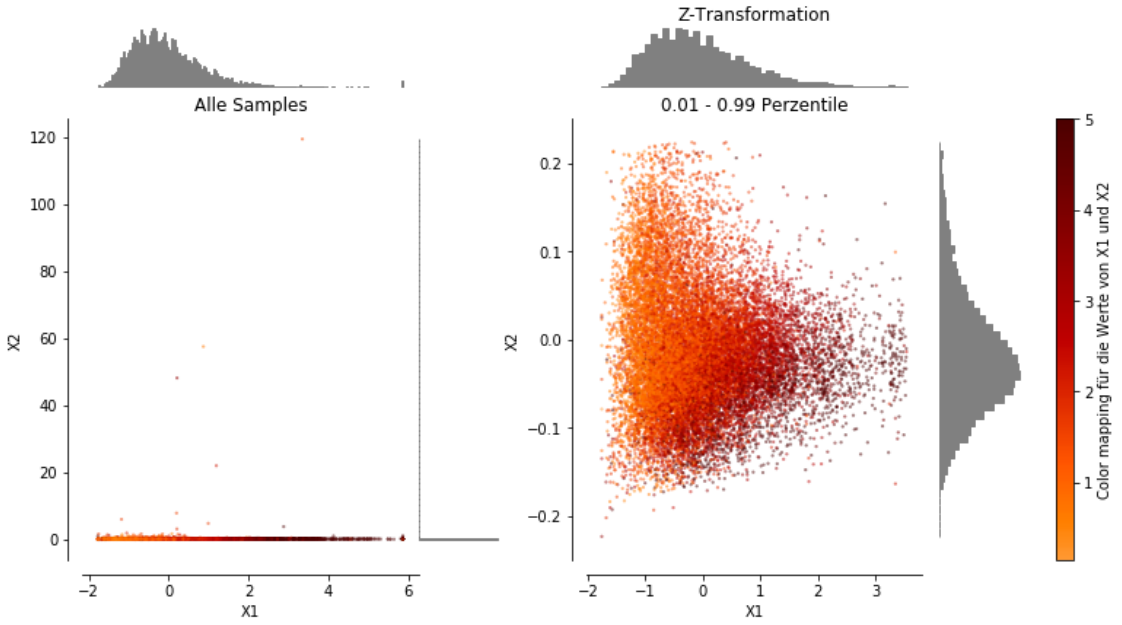


Abbildung 3.3: z-transformierter Datensatz

Quelle: nach (Raghav u. a., 2019)

In Abb. 3.3 ist anhand der Achsenskalierung von X2 der Einfluss der Ausreißer zu erkennen. Zwar wurde auf eine kleinere Größenordnung skaliert, jedoch wurde keine Gleichwertigkeit der Hauptmenge von X1 und X2 erreicht. Der Großteil von X2 ist in einen Bereich $[-0.2, 0.2]$ gepresst, während sich X1 im Bereich $[-2 - 3]$ befindet. X1 würde gegenüber der Hauptmenge

[18] entspricht dem arithmetischen Mittelwert
[19] Bamberg u. a., 2017.

von X2 in der Clusteranalyse stärker gewichtet, während die Ausreißer von X2 die Daten verzerren. Beides kann, durch ein vorheriges Entfernen der Ausreißer, verhindert werden.

3.3.2 Robuste Transformation

Um den Einfluss der Ausreißer beim Transformieren geringer zu halten, kann die robuste Transformation verwendet werden. Im Gegensatz zu Gleichung (3.5) wird in Gleichung (3.8) der Median $\tilde{x}$ subtrahiert und die Skalierung findet mit dem Interquartilsabstand[20] ($X_{75} - X_{25}$) statt. Die Achsenskalierung von X1 und X2 des rechten Diagramms in Abb. 3.4 sind nahezu gleich, obwohl die Verteilung des gesamten Datensatzes (links) mit dem der Ausgangsdaten übereinstimmt. Dadurch sind die Daten unterschiedlicher Feature gut vergleichbar, während die Information über die Ausreißer zugleich erhalten bleibt. Dies ermöglicht die Identifikation von zu hohen Werten, im Verhältnis zu den anderen Features.[21]

$$Z = \frac{X}{x_{75} - x_{25}} - \tilde{x} \tag{3.8}$$

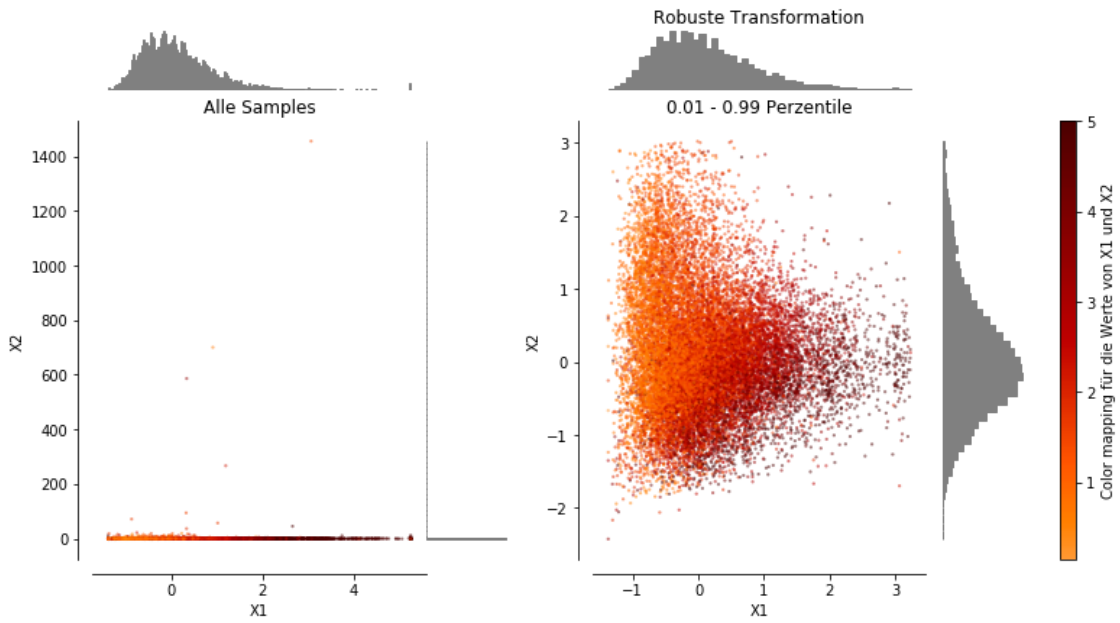


Abbildung 3.4: Robust transformierter Datensatzes

Quelle: (Raghav u. a., 2019)

[20] die Intervallbreite der mittleren 50% wenn die Werte des Features der Größe nach sortiert werden
[21] Raghav u. a., 2019.

3.3.3 Min-Max-Transformation

Bei der Min-Max-Transformation werden mit der Gleichung (3.9) alle Werte von X auf einen bestimmten Bereich $[min, max]$ normalisiert. Dabei wird der Vektor durch die Extremwerte x_{min} und $_{max}$ dividiert.

$$Z = \frac{X - x_{min}}{x_{max} - x_{min}} \cdot (max - min) + min \tag{3.9}$$

Der Einfluss der Ausreißer ist in Abb. 3.5 noch stärker als in Abb. 3.3 zu beobachten, da alle Daten in den Bereich $[0, 1]$ skaliert werden.[22] Jedoch ist der absolute Skalenbereich für alle Features identisch und somit lässt sich die Verteilung der Daten gut darstellen. Das ist besonders bei der Auswertung der Clusteranalyse hilfreich wenn einzelne Features ins Verhältnis zueinander gesetzt werden sollen.

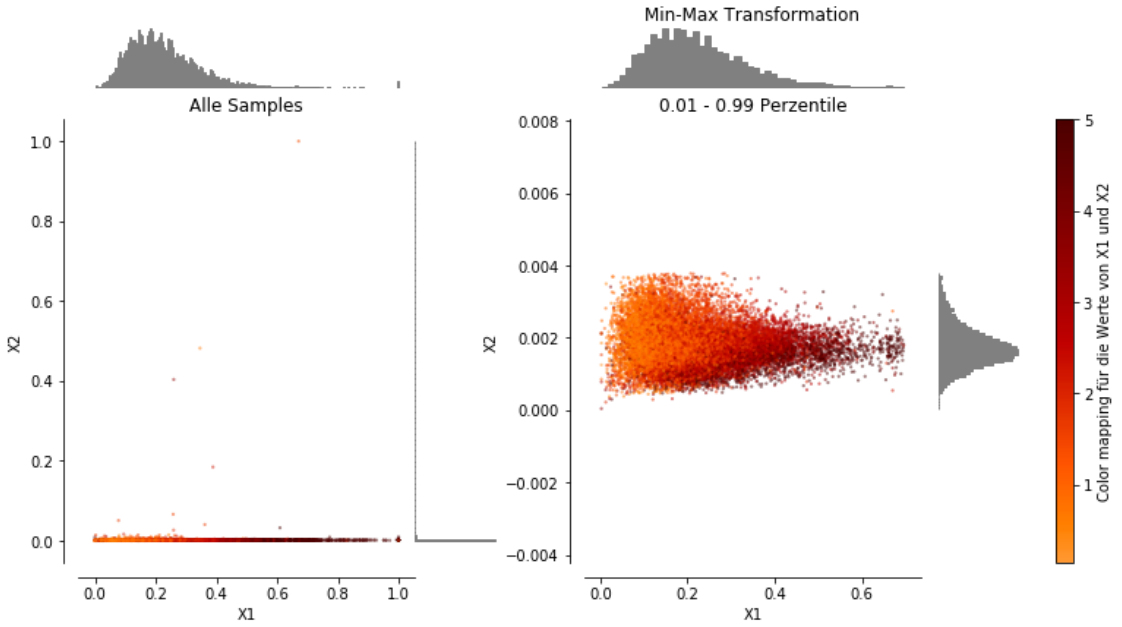


Abbildung 3.5: Min-Max-transformierter Datensatz

Quelle: (Raghav u. a., 2019)

[22] Raghav u. a., 2019.

3.4 K-Means

Wie schon in Abschnitt 2.3.3 angesprochen, ist K-Means ein häufig genutzter und validierter Cluster-Algorithmus. Seine Implementation in "Sci-kit learn„ wird in diesem Modell verwendet. Im wesentlichen, wird bei dem Verfahren die Intra-Cluster-Varianz $Comp(C)$ wie in Gleichung (3.10) formuliert, in der Zielfunktion minimiert.[23]

$$Comp(C) = \sum_{i=0}^{n} \min_{\bar{c_j} \in C} (\parallel x_i - \bar{c_j} \parallel^2) \tag{3.10}$$

Der Algorithmus teilt sich auf in zwei Schritte:

1. Zuordnung der Samples zum nächsten Cluster-Schwerpunkt
2. Aktualisieren der Cluster-Schwerpunkte

In Punkt 1 wird mit Gleichung (3.11) jedem Sample x_i der nächste Cluster-Schwerpunkt $\bar{c_k}$ über die Zuordnung $\hat{k}^{(k)}$ [24] mit der kleinsten euklidischen Distanz zugewiesen. Sollten zwei Schwerpunkte denselben Abstand haben, wird das Cluster mit der geringeren Anzahl an Zuweisungen gewählt.[25]

$$\hat{k}^{(i)} = \underset{k}{\operatorname{argmin}} \{dist_2(x_i - \bar{c_j})\} \tag{3.11}$$

In Punkt 2 wird dann der neue Cluster-Schwerpunkt $\bar{c_k}$ mit Gleichung (3.12) aus allen Samples eines Clusters $x_i(\hat{k})$ berechnet.

$$\bar{c_k} = \frac{\sum_i x_i(\hat{k})}{\mid x_i(\hat{k}) \mid} \tag{3.12}$$

Punkt 1 und Punkt 2 wiederholen sich so lange bis die Zuordnungen $\hat{k}^{(k)}$ sich nicht mehr ändern. Um das Verfahren starten zu können, ist eine initiale Cluster-Schwerpunkt-Zuordnung

[23] Lloyd, 1982; MacQueen, 1967.
[24] Mit $\hat{k}$ als die Zuordnung zum Cluster und k als die Anzahl der Cluster
[25] MacKay, 2003, S. 286.

notwendig. Hierbei werden einmalig zufällige Punkte im Raum ausgesucht. Dies hat die Folge, dass die Clusterzuweisungen bei jeder Anwendung verschieden sein können. In Abb. 3.6 sind die ersten vier Iterationsschritte dargestellt:

Iteration #0 Initialisierung und Zuweisung der Cluster

Iteration #1 Aktualisierung der Cluster-Schwerpunkte[26]

Iteration #2 Aktuallisierung und Neuzuordnung der Cluster[27]

Iteration #3 Aktualisierung und Neuzuordnung der Cluster

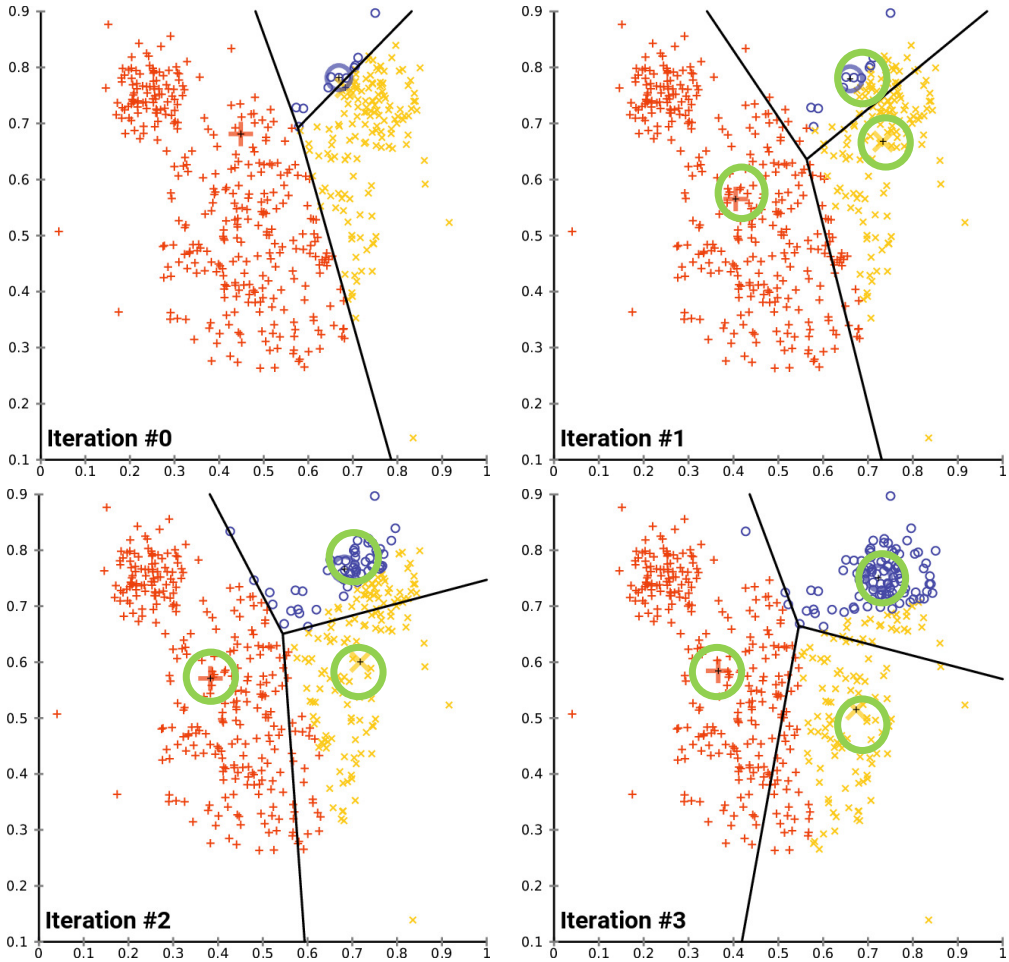


Abbildung 3.6: Visualisierung der ersten 4 Iterationsschritten des K-Means Algoritmus

Quelle: nach (Chire, 2017) CC BY-SA 4.0

Ein Nachteil von K-Means besteht darin, dass er sich in lokalen Minima verfangen kann. Deshalb sollte das Verfahren mehrmals mit unterschiedlichen Startpunkten gestartet werden und schließt das Setzen von eigenen Startpunkten quasi aus. Ein weiterer Nachteil ist die,

[26] diese sind mit grünen Kreisen hervorgehoben

[27] die farbliche Zuordnung bezieht sich hier noch auf die alte Zuweisung, während die Linen die neuen Clustergrenzen aufzeigen

durch die Intra-Cluster-Varianz bedingte Annahme, dass die Objekte innerhalb der Cluster kugelförmig verteilt sind. Liegt eine schiefe Verteilung oder unregelmäßige Formen vor, reagiert das Kompaktheitsmaß $Comp(C)$, das auf der euklidischen Distanz beruht, schlecht und die Cluster werden unsauber voneinander getrennt.

3.5 Validierungsindizes

Bei harten partitionierenden Clusterverfahren wie K-Means ist die Bestimmung der Anzahl der Cluster k nach denen gesucht werden sollen, der ausschlaggebendste Parameter. Gleichzeitig stellt die Bestimmung dieses Wertes auch die vermutlich schwierigste Aufgabe bei der Anwendung des Algorithmus dar. Aufgrund der Vielzahl von Iterationen, sind die Zwischenschritte nicht nachvollziehbar. Auch die Visualisierung von Daten, einer Dimension $>$ 3, ist ohne Komplexitätsreduktion nicht effizient möglich. Die Auswertung ist somit auf die Ergebnisse beschränkt.[28] Hierfür wurden zahlreiche Indizes entwickelt. Bei unüberwachten Methoden ist keine „richtige“ Clustereinteilung[29] zum Vergleichen vorhanden. Damit fallen externe oder vergleichende Indizes weg. Bei internen Indizes werden nur die Informationen genutzt, die durch die Daten selbst und die Clustereinteilung gegeben sind. Die Cluster werden anhand der Kompaktheit $Comp$[30] und der Trennung Sep[31] voneinander bewertet. Die Kompaktheit $Comp$ wird oft mit Hilfe der Varianz oder der Distanz aller Samples im Cluster berechnet. Bei der Trennung Sep hingegen oft mit paarweiser Distanzen von Clusterschwerpunkten oder minimaler Distanzen zwischen zwei Clustern. Arbelaitz u. a.[32] sowie Milligan und Cooper[33] haben viele Indizes miteinander verglichen und ausgewertet. Drei optimierenden Indizes wurden für dieses Modell ausgewählt. Gut voneinander getrennte aber dichte Cluster führen zu einer besseren Bewertung. Je nach Index, wird der optimale Punkt als Maximum oder Minimum identifiziert.

[28] Halkidi u. a., 2001, 16 f.
[29] oft *eng.* Groundtruth
[30] Im Folgenden als Variable $Comp$ abgekürzt
[31] Im Folgenden als Variable Sep abgekürzt
[32] Arbelaitz u. a., 2013.
[33] Milligan und Cooper, 1985.

3.5.1 Calinski-Harabasz-Index

Der Calinski-Harabasz-Index CH beschreibt das Verhältnis von „Trennung der Cluster“ zu „Kompaktheit innerhalb der Cluster“. Die beste Clustereinteilung wird durch einen maximalen Wert beschrieben. Wie aus Gleichung (3.13) ersichtlich, wird der Zähler maximiert, während der Nenner minimiert werden sollte. Der Faktor $\frac{N-K}{K-1}$ verhindert mit der Anzahl an N Samples im Set und der Anzahl an K gefundenen Clustern, dass der Quotient monoton mit der Clusteranzahl wächst.[34]

$$CH = \frac{Sep(C)}{Comp(C)} \times \frac{N-K}{K-1} \tag{3.13}$$

Der Index schätzt die $Comp(C)$[35] in Gleichung (3.14) durch die Quadratsumme des Abstands der Samples zu ihrem Cluster-Schwerpunkt $\bar{c_k}$ ab[36], die in Gleichung (3.15) formuliert ist.[37]

$$Comp(C) = \sum_{c_k \in C} Comp(c_k) \tag{3.14}$$

$$Comp(c_k) = \sum_{x_i \in c_k} dist_2(x_i - \bar{c_k})^2 \tag{3.15}$$

Die $Sep(C)$[38] wird in Gleichung (3.16) durch die Summe der Distanzen der Cluster-Schwerpunkte $\bar{c_k}$ zum globalen Schwerpunkt $\bar{X}$ definiert, die auch als Inter-Cluster-Varianz bekannt ist (Gleichung (3.17))

$$Sep(C) = \sum_{c_k \in C} Sep(c_k) \tag{3.16}$$

$$Sep(c_k) = \mid c_k \mid (\bar{c_k} - \bar{X})^2 \tag{3.17}$$

[34] Vendramin u. a., 2010.
[35] Kompaktheit
[36] auch Inter-Cluster-Varianz genannt
[37] Caliński und Harabasz, 1974.
[38] Trennung

Nach Milligan und Cooper[39] erzielt der Calinski-Harabasz-Index die besten Ergebnisse. Auch wegen diesem Paper ist er ein häufig genutzter Index. Liu u. a.[40] merkt hierzu an, dass der Index empfindlich auf Rauschen reagiert, da Gleichung (3.15) schneller ansteigt als Gleichung (3.17) und die resultierenden Werte somit abnehmen.[41] Wie schon in Abschnitt 3.4 erwähnt, setzt die Intra-Cluster-Varianz eine kugelförmige Verteilung um den Schwerpunkt eines Clusters voraus. Bei Abweichungen von dieser Annahme verliert der Index an Genauigkeit.

3.5.2 Davis-Bouldin-Index

Ein weiterer viel genutzter Index, ist der von Davies und Bouldin DB. Dieser Index berechnet den gemittelten Quotienten von Kompaktheit und Trennung zweier Clusterpaare. $Comp(c_k)$ und $Comp(c_l)$ des Paares werden addiert und durch die Trennung der Clusterpaare voneinander $Sep(c_k, c_l)$ dividiert. Nach Gleichung (3.18) werden die jeweils größten Werte aufsummiert und gemittelt.

$$DB = \frac{1}{K} \sum_{c_k \in C} \max_{c_l \in C \neq c_k} \left\{ \frac{Comp(c_k) + Comp(c_l)}{Sep(c_k, c_l)} \right\} \tag{3.18}$$

Hierbei wird $Comp(c)$ nach Gleichung (3.19) als mittlere euklidische Distanz zwischen Sample x_i und Cluster-Schwerpunkt $\bar{c_k}$ sowie $Sep(c_k, c_l)$ nach Gleichung (3.20) als euklidische Distanz zweier Cluster-Schwerpunkte $\bar{c_k}$, $\bar{c_l}$ berechnet.

$$Comp(c_k) = \frac{1}{\mid c_k \mid} \sum_{x_i \in c_k} dist_2(x_i, \bar{c_k}) \tag{3.19}$$

$$Sep(c_k, c_l) = dist_2(\bar{c_k}, \bar{c_l}) \tag{3.20}$$

Somit wird ein Gütemaß für jedes Cluster berechnet und alle miteinander verglichen. Nur der jeweils höchste Wert geht in die Wertung ein. Deshalb kann der Wert bei sehr nahen

[39] Milligan und Cooper, 1985.
[40] Liu u. a., 2010.
[41] Liu u. a., 2010.

Clustern ungenau sein.[42] Diese Werte werden dann gemittelt. Um Cluster zu erhalten, die sich möglichst deutlich voneinander unterschieden, sollte der Index minimiert werden.

3.5.3 Silhouetten-Index

Von Rousseeuw[43] definiert, ist der Silhouetten-Index Sil ein normiertes Maß zur Beschreibung der Clustergüte. Dieser Index beschreibt die Zuordnung eines Samples zu seinem Cluster (Kompaktheit) in Abhängigkeit der Distanz zum nächsten Cluster (Trennung). Charakteristisch dafür, stehen höhere Werte von $s \in [-1, 1]$ für eine höhere Güte. Der Wert $s = 0$ steht für eine uneindeutige Zuordnung oder sich überschneidende Cluster. Eine klare Falschzuordnung würde sich durch $s = -1$ ausdrücken. Die Berechnung der Differenz aus Kompaktheit und Trennung erfolgt für jedes Sample und wird dann gemittelt. Die Normierung erhält der Index, da er durch das Maximum von Kompaktheit $Comp(C)$ und Trennung $Sep(C)$ geteilt wird. Die Gleichung lautet:

$$Sil = \frac{1}{N} \sum_{c_k \in C} \sum_{x_i \in c_k} \frac{Sep(x_i, c_k) - Comp(x_i, c_k)}{\max\left(Comp(x_i, c_k), Sep(x_i, c_k)\right)} \tag{3.21}$$

mit $Comp(x_i, c_k)$ als die mittlere Distanz zwischen einem Sample x_i und allen Samples c_i im selben Cluster wie in Gleichung (3.22).

$$Comp(x_i, c_k) = \frac{1}{\mid c_k \mid} \sum_{x_i \in c_k} dist_2(x_i, x_k) \tag{3.22}$$

und $Sep(x_i, c_k)$ als kleinster Wert der mittlere Distanz des selben Samples x_i zu allen Samples c_k des nächsten Clusters wie in Gleichung (3.23).

$$Sep(x_i, c_k) = \min_{c_l \in C \neq c_k} \left\{ \frac{1}{\mid c_l \mid} \sum_{x_j \in c_l} dist_2(x_i, x_j) \right\} \tag{3.23}$$

[42] Liu u. a., 2010.
[43] Rousseeuw, 1987.

Abb. 3.7 veranschaulicht diese Berechnung. Zuerst wird die mittlere Distanz $Comp(x_i, c_k)$ im eigenen Cluster A berechnet, anschließend $Sep(x_i, c_k)$ aus allen anderen Clustern B und C. Das Cluster mit dem minimalen Wert geht dann als b in Gleichung (3.21) ein.

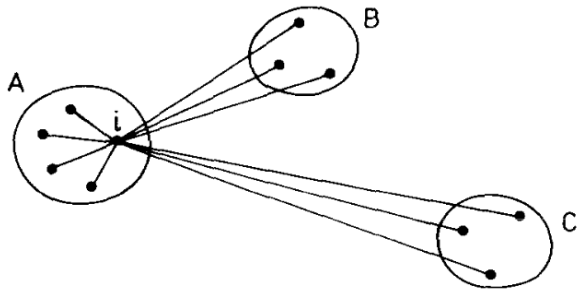


Abbildung 3.7: Darstellung der Berechnung des Silhouetten-Index

Quelle: (Rousseeuw, 1987)

Die Wertung basiert somit auf der paarweisen Differenz der Abstände zwischen und innerhalb der Cluster. Da die Werte des Silhouetten-Index zwischen den Grenzen -1 und +1 liegen, erleichtert er den Vergleich der Ergebnisse verschiedener Parameter-Sets als gemittelter Wert. Aber die einzelnen Cluster lassen sich mit den Werten der Samples miteinander vergleichen.

4 Modellerstellung

Das Modell wurde objektorientiert in der Sprache Python programmiert und ist für die Anwendung in einem „Jupyter Notebook“ ausgelegt. Es bietet Funktionen zum durchführen, vergleichen und auswerten von Clusteranalysen mit dem Algoritmus K-Means. Um das Vergleichen einzelner Schritte zu vereinfachen wurde eine Datenstruktur entwickelt, in der Zwischenergebnisse gespeichert und abgerufen werden können. Die im Modell genutzten Daten entsprechen der NUTS-3 Ebene. Für die Zuordnung der NUTS-3 Codes ist eine Transfertabelle[1] notwendig. Diese wird für Daten ab NUTS 2013 (gültig ab 2015) mit dem Modell bereitgestellt[2]. Die wesentlichen Pakete die im Modell genutzt wurden sind: Scikit-learn, Geopandas, Seaborn und Matplotlib. Eine vollständige Liste der notwendigen Pakete ist in der Datei enviroment.yml zu finden. Eine kurze Anleitung, wie diese zum Erstellen eines Environment unter Anaconda genutzt wird, befindet sich in der Datei „README.rtf“. Beides wird in digitaler Form übergeben und ist auch auf Github[3] zu finden.

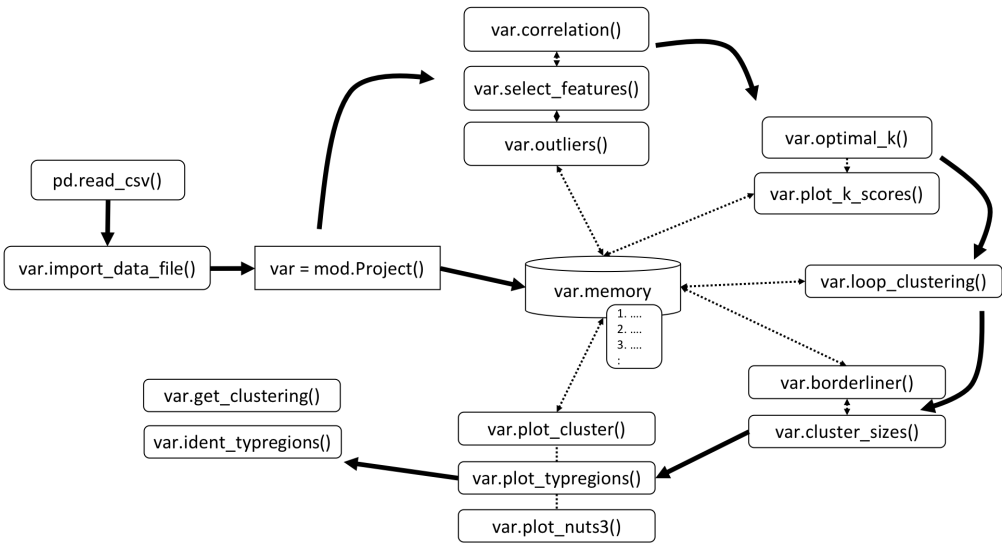


Abbildung 4.1: Schematischer Modellablauf

Quelle: eigene Darstellung

In Abschnitt 4.1 wird zuerst auf die Datenstruktur eingegangen und danach einzelne Funktion entsprechend der ungefähren Modellreihenfolge erklärt. Diese ist in Abb. 4.1 schematisch

[1] Änderungen können unter folgendem Link nachvollzogen werden https://ec.europa.eu/eurostat/de/web/nuts/history

[2] erstellt aus eurostat, 2018b; Destatis, 2017

[3] https://github.com/nailend/ModITy.git

durch die starken Striche dargestellt. Die gestrichelten Pfeile stellen den Austausch mit dem Speicher dar. Da es sich um ein iteratives-Verfahren handelt müssen Parameter für jeden Speichersatz einzeln definiert werden. Im folgenden Text sind die Parameter der Modellfunktionen in Farbe. Die Farbgebung lässt hierbei auf den Typ schließen und soll den Gebrauch erleichtern. `Variablen` können beliebig benannt werden. `Funktionen()` beinhalten meist mehrere Parameter wie `numerics/boolean` oder `strings`. `Default-values` sind im Constructor voreingestellt und müssen nur bei einer Abweichung übergeben werden.

4.1 Datenstruktur

Um die Ergebnisse bei Parametereinstellungen vergleichbar zu machen, ist ein Speichern der jeweils gesetzten Parameter notwendig. Diese werden für jede getroffene Konstellation in dem Klassen-Attribut `memory` chronologisch gespeichert. Cluster-Zuweisungen sowie andere Endergebnisse werden ebenfalls abgespeichert. Ein Speichersatz beinhaltet die in Tabelle 4.1 aufgelisteten Informationen und ist selber als Typ dictionary implementiert.

keyword	**type**
samples	dict
features	dict
outlier	tuple
scores	pd.DataFrame
labels	pd.DataFrame
scaler	str

Tabelle 4.1: Struktur eines Speichersatzes

4.2 Einleseroutine

```
import modity.py as mod
import pandas as pd
```

Zuerst wird das Pandas Paket und das Modell eingebunden:

```
bachelor=mod.Project(input_folder=string)
```

Die Klasse `Project` kann in einer beliebigen Variable (bspw. `bachelor`) instanziiert werden. Dabei muss der Eingangsdatenordner als Parameter übergeben werden. Daraufhin wird der Constructor initiiert und alle notwendigen Attribute und Ordnerstrukturen (graphics, output) werden erstellt. Außerdem wird der Ordner nach csv-Dateien durchsucht und diese mit

Namen in einer Tabelle angezeigt:

`pd.read_csv()`

Die Eingangsdaten müssen mit einer Importfunktionen des Pandas Paketes als `pd.DataFrame` eingelesen und in einer Variable (bspw. `file`) gespeichert werden. Es ist wichtig, auf die Codierung `encoding`, die Trennzeichen `sep`, Tausendertrennzeichen `thousands` und Dezimalzeichen `decimal` zu achten. Alle Werte müssen entweder als Integer `int` oder Gleitkommazahl `float` erkannt werden. Der Eingangsdatensatz kann dann, bis auf die gewollten Features sowie die Regionalcodespalte bereinigt werden. Die Kopfzeile der Regionalcodespalte sollte ihrem Typ entsprechend (AGS, PLZ oder NUTS3)[4] werden. Anschließend wird die Variable `file` mit dem `pd.DataFrame` unter Angabe des Regionalcodes an die Klasse übergeben:

```
bachelor.import_data_file(file=file, code_type=string)
        code_type: 'AGS','PLZ','NUTS3'
```

Den Daten werden die NUTS3-Codes zugeordnet und als Klassenattribut unter `bachelor.df_data` gespeichert. Der Speicher `bachelor.memory` wird initiiert und ab sofort für alle Funktionen genutzt. Der komplette Datensatz wird als nullter Eintrag hinterlegt[5].

4.3 Datenauswahl

Vor der Anwendung des Clusteralgorithmus sollten die Daten weiter selektiert werden damit ein interpretierbares Ergebnis erzielt werden kann. Korrelierende Features lassen sich mit der Funktion aus Abschnitt 4.3.1 identifizieren. Abschnitt 4.3.2 zeigt, wie Feature von Speichersätzen entfernt bzw. zugefügt werden können. Funktionen in Abschnitt 4.3.3 identifiziert mit der robusten Transformation Ausreißer und entfernen diese bzw. spezifische Samples vom Speichersatz. Einen Überblick über deren Zusammenstellung verschaffen die Funktionen aus Abschnitt 4.3.4

4.3.1 Korrelationsanalyse

```
bachelor.correlation(method=string, memory_set=None,
                     heatmap=False, threshold=None, save=False)
        method: 'pearson', 'kendall', 'spearmann'
```

[4] Mehrere Datensätze unteschiedlicher Regionalcodes können nicht verarbeitet werden
[5] `memory_set=0`

Mit dieser Funktion werden korrelierende Features identifiziert. Der Parameter `method` bestimmt den Koeffizienten (Pearson, Kendall oder Spearmann). Regulär wird der neuste Speichersatz ausgewählt, falls kein Wert für `memory_set` übergeben wird. Mit `heatmap` werden die Koeffizienten in einer Heatmap dargestellt und `save` wird diese im graphics-Ordner abgespeichern. An `threshold` kann ein Grenzwert übergeben werden. Feature-Paare, die diesen überschreiten, werden in einer Liste ausgegeben. Da jeweils nur ein Feature der korrelierenden Feature-Paare entfernt werden muss, wurde der Prozess nicht automatisiert. Analysierende müssen diese Entscheidung eigenständig treffen. Mit der Feature-Selection Funktion sind diese zu entfernen.

4.3.2 Feature-Selection

```
bachelor.select_feature(memory_set=0, add=[int], reject=[int])
```

Mit Hilfe der `select_feature` Funktion kann eine Untermenge des Rohdatensatzes erzeugt und im Speicher hinterlegt werden. Über `memory_set` wird der Ausgangsdatensatz gewählt. Von diesem werden mit `add` bzw. `reject` einzelne oder mehrere Features wieder hinzugefügt bzw. entfernt. Der Parameter `add` wird hierbei bevorzugt behandelt, falls ein Wert in beide Parameter übergeben wird. Mit jeder Ausführung wird ein neuer Speichereintrag mit aufsteigender Nummerierung erstellt.

4.3.3 Ausreißer Identifikation

```
bachelor.outliers(memory_set=0, save=False)
```

Diese Funktion bietet die Möglichkeit Ausreißer zu identifizieren und von einem neuen Speichersatz zu entfernen. Über den Parameter `memory_set` wird ein beliebiger Speichersatz[6] als Basis ausgewählt. Dieser wird robust transformiert, damit die Features in Relation gesetzt werden können. Ein Diagramm veranschaulicht die Werte. Der Parameter `save` ermöglicht es den Graphen, inklusive eingetragener Grenzwertlinie im PDF-Format zu speichern. Aufgrund der Transformation sind die Werte nur qualitativ zu interpretieren. Über eine Eingabeaufforderung wird ein positiver und negativer Grenzwert abgefragt. Samples die diesen für bestimmte Features überschreiten werden identifiziert und in einer Tabelle ausgegeben.

[6] der Rohdatensatz hat den Wert `0`

Durch eine weitere Eingabeaufforderung kann das Entfernen der Ausreißer bestätigt werden. Falls dies geschieht wird die Untermenge als neuer Eintrag im Speicher abgelegt.

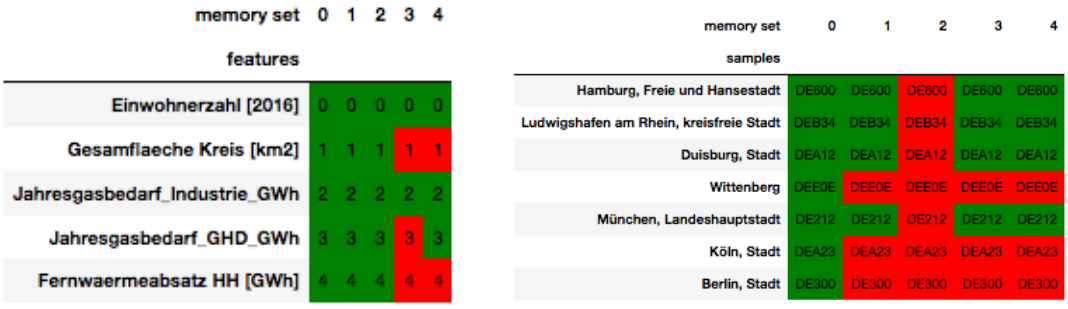
```
bachelor.select_samples(memory_set=0, add=[string], reject=[string])
```

Auch einzelne bzw. mehrere Samples können anhand ihres NUTS-3 Code von einem Speichersatz entfernt bzw. zugefügt werden. Der neue Speichersatz wird dann abgespeichert. Hierfür müssen die NUTS-3 Codes als string oder als Liste von strings in `add` bzw. `reject` übergeben werden.

4.3.4 Vergleich

`bachelor.show_features()` `bachelor.show_samples()`

Um einen Überblick über die Feature-Sample Konstellationen der einzelnen Speichersätze zu erhalten, können die hier genannten Funktionen genutzt werden. Dieser wird wie in Abb. 4.2 tabellarisch dargestellt. Rote Werte stellen hierbei ausgeschlossenen und grüne vorhandene Features bzw. Samples dar.



memory set	0	1	2	3	4
features					
Einwohnerzahl [2016]	0	0	0	0	0
Gesamflaeche Kreis [km2]	1	1	1	1	1
Jahresgasbedarf_Industrie_GWh	2	2	2	2	2
Jahresgasbedarf_GHD_GWh	3	3	3	3	3
Fernwaermeabsatz HH [GWh]	4	4	4	4	4

(a) show_features()

memory set	0	1	2	3	4
samples					
Hamburg, Freie und Hansestadt	DE600	DE600	DE600	DE600	DE600
Ludwigshafen am Rhein, kreisfreie Stadt	DEB34	DEB34	DEB34	DEB34	DEB34
Duisburg, Stadt	DEA12	DEA12	DEA12	DEA12	DEA12
Wittenberg	DEE0E	DEE0E	DEE0E	DEE0E	DEE0E
München, Landeshauptstadt	DE212	DE212	DE212	DE212	DE212
Köln, Stadt	DEA23	DEA23	DEA23	DEA23	DEA23
Berlin, Stadt	DE300	DE300	DE300	DE300	DE300

(b) Vergleich der ausgeschlossenen Samples je Set

Abbildung 4.2: Beispielhafte Darstellung der Vergleichs-Funktionen

Quelle: eigene Darstellung

4.4 Clustern

Das eigentliche Cluster findet mit K-Means und einem transformierten Datensatz statt. Das Transformationsverfahren sowie die Clusteranzahl werden als Parameter vorgegeben. Hierfür werden die Funktionen aus Abschnitt 4.4.1 genutzt. Um die Parameter zu optimieren, werden diese variiert und die resultierenden Cluster mit den Validierungsindizes überprüft

und verglichen. Abschnitt 4.4.2 beschreibt wie die Ergebnisse, eines bestimmten Parametersatzes, noch tiefgehender untersucht werden können. Zuletzt werden in Abschnitt 4.4.3 die Typregionen identifiziert.

4.4.1 Bestimmung der Clusteranzahl

```
bachelor.optimal_k(memory_set=int, kmin=2, kmax=int
                        scaler='standard', plot=True, new=False)
          scaler: 'standard', 'minmax', 'robust
```

Der mit `memory_set` ausgewählte Speichersatz wird mit dem in `scaler` ausgewählten Transformationsverfahren, transformiert. Anschließend iteriert der Algorithmus K-Means über alle Werte von k im Intervall von `kmin` bis `kmax` und berechnet die Validierungsindizes aus Abschnitt 3.5. Die Indizes werden in Tabellenform in `memory` gespeichert und können mit `plot` als Liniendiagramme dargestellt werden. Dies ermöglicht den direkten Vergleich unterschiedlicher k-Werte. Die Indizes sind wie in Abschnitt 3.5 beschrieben, nach Maximum bzw. Minimum zu optimieren. Bei der Methode der kleinsten Fehlerquadrate sucht man nach einem Knick im Verlauf. Die mathematisch optimalen Wert werden für jeden Wert markiert, sind jedoch nur ein Indiz und dienen lediglich zur Orientierung. Die Plots können für unterschiedliche Werte der `memory_sets`, `scaler` und `kmax` erstellt werden. Wurden die Indizes einmal berechnet, sind sie auch mit unterschiedlichen Speichersätzen vergleichbar. Jedoch sind nur der Silhouetten-Index und der Davis-Bouldin-Index normiert und bei unterschiedlicher Sample/Feature-Konstellation aussagekräftig.

```
bachelor.plot_k_scores(memory_set=int, save=False)
```

Über `memory_set` kann eine Liste von Speichersatznummern übergeben werden. Mit `save` werden die Grafiken im PDF-Format gespeichert.

4.4.2 Parameterprüfung

Wurde ein Speichersatz oder mehrere als vielversprechend identifiziert, können die Parameter noch genauer überprüft werden. Da sich K-Means in lokalen Minima verfangen kann, sollte der Algorithmus mehrmals mit den gleichen Parameter gestartet werden. Hierbei können einzelne Samples entfernt werden, um eine robustere Zuweisung zu erhalten. Das

memory_set und eine Clusteranzahl k_cluster müssen übergeben werden. Die Transformationsmethode wird aus dem Speichersatz übernommen. Die Anzahl der Wiederholungen kann mit max_loops in der folgenden Funktion definiert werden:

```
bachelor.loop_clustering(memory_set=int, k_cluster=int, max_loops=int)
```

Dabei werden die Indizes berechnet und die Clustezuweisungen jeder Iteration gespeichert. Wie schon in Abschnitt 3.4 erwähnt, sind aufgrund der Initialisierung des Algorithmus die Clusterbenennungen jedes mal anders. Um diese dennoch vergleichen zu können, werden die Benennungen an den ersten Lauf angepasst. Dabei kann es zu Fehlzuweisungen der Cluster kommen, über die eine Ausgabe informiert. Die entsprechenden Iterationen sind deshalb mit besonderer Sorgfalt zu betrachten.

```
bachelor.borderliner(memory_set=int, k_cluster=int, exclude=None)
```

Mit dieser Funktion können die Samples identifiziert werden, die während den Iterationen unterschiedlichen Clustern zugewiesen wurden. Die Parameter memory_set und k_cluster sollten wie bei der vorherigen Funktion eingestellt werden. Mit exclude können die Iterationen mit evtl. Fehlzuweisungen ausgenommen werden. Die identifizierten Samples werden als Tabelle ausgegeben und im Speicher hinterlegt. Außerdem wird eine Übersicht erstellt, die genauere Informationen (Name, Clusterzugehörigkeiten) enthält. Wenn es sich hierbei nur um eine kleine Gruppe handelt, ist zu überlegen ob diese kurzzeitig ausgeschlossen werden sollten, da das Lösungsverhalten von K-Means beeinflussen. Um sie nicht ganz aus der Lösungsmenge zu verlieren könnten sie nachträglich zugeordnet werden.

```
bachelor.cluster_sizes(memory_set=int, k_cluster=int)
```

Mit den identischen Parametern gibt diese Funktion eine Tabelle der Clustergrößen für jede Iterationen aus. Iterationen die identische Zuweisungen hatten werden aussortiert. Cluster mit sehr geringer Größe können genauer untersucht werden und die zugehörigen Samples gegebenenfalls aussortiert werden.

```
bachelor.plot_silhouettes(memory_set=int, k_cluster=int, loop=int,
                          save=False, output=False)
```

Mit dem Silhouette Plot kann eine spezifische Clusterung hinsichtlich der Zuordnung einzelnen Samples ausgewertet werden. Hierfür müssen die exakten Parameter memory_set, k_cluster und loop übergeben werden. Das Diagramm stellt die Silhouettenkoeffizienten jedes Samples für ihre Cluster dar. Haben Samples Werte kleiner null deutet dies auf eine falsche Zuordnung hin. Um diese spezifisch zu untersuchen, können die Werte mit dem

Parameter `output` ausgegeben werden.

4.4.3 Identifikation der Typregionen

```
bachelor.ident_typregions(memory_set=int, k_cluster=int, loop=int)
```

Zur Identifikation der Typregionen wird der exakte Speichersatz ausgewählt. `memory_set`, `k_cluster` und `loop` sind die bestimmenden Parameter. Wobei sich `loop` auf eine bestimmte Iteration aus der zuvor stattgefundenen Parameterprüfung bezieht. Die Typregionen werden aus dem geringsten Abstand zum jeweiligen Schwerpunkt der Cluster abgeleitet. Da sich das Clustering nur auf den ausgewählten Speichersatz bezieht, werden auch nur die entsprechenden Werte mit in der Tabelle ausgegeben.

4.5 Visualisierung und Datenausgabe

Auf alle Teilergebnisse kann über das `memory`-Klassenattribute zugegriffen werden. Hierbei ist auf Tabelle 4.1 verwiesen. Diese Struktur wird für jeden Speichersatz angelegt. Zusätzlich gibt es Funktionen, die einen bestimmten Datensatz reproduziert:

```
bachelor.get_dataset(memory_set=int)
```

Wurde ein Parametersatz, wie in Abschnitt 4.4.2 überprüft, ist es möglich den geclusterten Speichersatz bzw. nur die Clusterzuordnung auszugeben.

```
bachelor.get_clustering(memory_set=int, k_cluster=int, loop=int)
bachelor.get_labels(memory_set=int, k_cluster=int, loop=int)
```

Um die Endergebnisse in einen regionalen Kontext zu setzen, ermöglichen es drei Funktionen, das gesamte Clustering, die Typregionen oder eine beliebige Liste an NUTS-3 Codes in einer Deutschland-Karte, farblich differenziert darzustellen. Hierfür müssen die Parameter `memory_set`, `k_cluster` und `loop` der Clusterzuweisung übergeben werden. Mit `save` wird festgelegt ob die Grafiken im PDF-Format exportiert werden.

```
bachelor.plot_cluster(memory_set=int, k_cluster=int, loop=int, save=False)
bachelor.plot_typregions(memory_set=int, k_cluster=int, loop=int, save=False)
bachelor.plot_nuts3(nuts3, save=False)
```

5 Anwendung

Im Folgenden werden zwei mögliche Varianten, die mit dem Modell exemplarisch durchgeführt wurden, beschrieben. Hierbei werden schrittweise die Datensätze verfeinert bis die Zielvorgabe des Algorithmus eine eindeutige und robuste Lösung erzielt. Die Modelldurchführungen liegen der Abschlussarbeit digitale bei. Als Grundlage dient der bereitgestellte Datensatz des Projekts „AIRE “. Zuerst wird in Abschnitt 5.1 der Datensatz aufgrund hoher Korrelationswerten reduziert. In Abschnitt 5.2 und Abschnitt 5.3 werden daraufhin 2 unterschiedliche Varianten angewendet. Abschnitt 5.4 beschreibt und vergleicht die Ergebnisse. Abschließend werden diese und die Grenzen des Modells in Abschnitt 5.5 diskutiert.

5.1 Datenauswahl

Wie schon in Abschnitt 2.3.1 erwähnt, wird das Ergebnis bei einer Vielzahl von Features durch ausreißende Werte schnell verzerrt und uneindeutig. Da der Ausgangsdatensatz eine relativ kleinen Menge an Samples und hohe Anzahl an Features besitzt verstärkt sich dieser Effekt. Mit Blick auf die Korrelationsanalyse[1] in Abb. A.2a fallen starke Abhängigkeiten auf. Redundante Features mit hohen Werten werden entfernt, um eine gleichmäßige Gewichtung der Informationen zu gewährleisten. Die in Tabelle 5.1 aufgelisteten Features werden für die Clusteranalyse ausgewählt und sollen einen Querschnitt der Daten liefern. Ihre linearen Abhängigkeiten wurden weitgehend minimiert und sind in Abb. A.2b zu sehen. Diese Auswahl wird im Folgenden für beide Varianten genutzt.

Tabelle 5.1: Ausgewählte Features

Features	
Einwohnerdichte	Fernwärmeabsatz GHD
Jahresgasbedarf Haushalte	Siedlungsfläche
Geothermiepotenzial	Jahresgasbedarf Industrie

[1] Da es sich nicht um normalverteilte Variablen handelt wurde der Koeffizient nach Kendall gewählt.

5.2 Variante 1

Zuerst werden mögliche Ausreißer in den Daten von Speichersatz 1 identifiziert. Diese werden, wie in Abschnitt 3.3.2, robust transformiert und mit dem gewählten Grenzwert 15 entfernt. Eine Darstellung ist in Abb. A.1 zu sehen. Die Samples in Tabelle 5.2 wurden dadurch aus Speichersatz 2 entfernt. Der Einfluss auf die standardisierten Daten ist in Abb. A.3 zu sehen. In Abb. A.4a sind die Verläufe der Validierungsindizes für unterschiedliche Werte von k, für alle verwendeten Speichersätze in Linendiagrammen dargestellt. Die identifizierten Extrema befinden sich in Tabelle 5.3. Für den in Speichersatz 2, mehrheitlich empfohlenen Wert k=7 werden bei 30 Iterationen robustere[2] Clustergrößen erlangt als bei k=6. Diese sind in Tabelle 5.5 zu sehen[3]. Dabei sticht Cluster 4 als „Singleton“[4] heraus. Da sehr kleine Cluster nicht aussagekräftig sind, wurde das Sample entfernt. Die Indizes in Tabelle 5.5 deuten bei dem neu erzeugten Speichersatz 3 mehrheitlich auf die Clusteranzahl k=6. Das ist auch in Abb. A.4a nachzuvollziehen. Hiermit werden noch robustere Ergebnisse als zuvor erreicht. Aus den Zuweisungen der Iterationen wurden die Clusterspringer in Abb. A.4a identifiziert. Testweise werden diese entfernt. Beim erneuten Iterieren mit Speichersatz 4 wurde ein eindeutiges Ergebnis erlangt.

Tabelle 5.2: Ausreißer Variante 1

NUTS3	**Name**
DE712	Frankfurt am Main, Stadt
DEE0E	Wittenberg
DE212	München, Landeshauptstadt
DEA12	Duisburg, Stadt
DE300	Berlin, Stadt
DED21	Dresden, Stadt
DE600	Hamburg, Freie und Hansestadt
DED51	Leipzig, Stadt
DEB34	Ludwigshafen am Rhein, kreisfreie Stadt
DE254	Nürnberg
DEA23	Köln, Stadt

[2]weniger unterschiedliche Clusterzuordnungen und weniger variierende Clustergrößen
[3]Iterationen mit der selben Clusterzuweisung wurden entfernt
[4]Ein Cluster mit nur einem Sample

Tabelle 5.3: Validierungsindizes für Speichersatz 2 ,3 und 4

	Speichersatz		
Index	**2**	**3**	**4**
kleinste Fehlerquardrate SSR	4	4	4
Silhouetten-Index Sil	6	6	6
Davis-Bouldin-Index DB	7	6	6
Calinski-Harabasz-Index CH	7	6	6

Tabelle 5.4: Ergebnisse der Iteration von Speichersatz 2 für k=7

Clustergrößen

	Loop						
cluster	0	1	2	7	9	10	11
0	47	47	46	47	47	47	46
1	69	68	68	69	69	69	68
2	23	23	22	23	19	20	22
3	82	83	82	83	81	83	83
4	1	1	1	1	6	1	1
5	10	10	10	10	9	10	10
6	158	158	161	157	159	160	160

Sample aus Cluster 4

NUTS3	Name
DE929	Region Hannover

Tabelle 5.5: Ergebnisse der Iteration von Speichersatz 3 für k=6

Clustergrößen

	Loop					
cluster	0	1	2	5	15	25
0	158	160	158	159	159	157
1	85	85	83	85	83	86
2	23	22	23	22	23	21
3	10	10	10	10	10	10
4	47	46	47	47	47	48
5	66	66	68	66	67	67

Springende Sampels

NUTS-3	**Name**
DEG05	Weimar, Stadt
DE244	Hof
DEF0C	Schleswig-Flensburg
DEF0B	Rendsburg-Eckernförde
DEB35	Mainz, kreisfreie Stadt
DEA52	Dortmund, Stadt
DE216	Bad Tölz-Wolfratshausen
DE945	Wilhelmshaven, Stadt

5.3 Variante 2

Im Gegensatz zu Abschnitt 5.2 wird die Identifikation der Ausreißer in Variante 2 nicht mit der Methode der robusten Transformation vorgenommen. Dementsprechend wird der reduzierte Datensatz aus Abschnitt 5.1 direkt mit Unterstützung der Validierungsindizes und variierender Clusteranzahl betrachtet. Die Indizes in Abb. A.5 lassen jedoch keinen eindeutigen

Schluss auf eine bestimmte Clusteranzahl zu. Allerdings lässt der sehr hohe Silhoutten-index für k=2 auf Ausreißer deuten. Um sinnvolle Indexwerte zu erlangen, müssen diese entfernt werden. Folge dessen wurde der Speichersatz 1 mit k=2 iteriert. Die Ergebnisse sind in Tabelle 5.6 zu sehen. Bei 30 Iterationen bildeten sich 2 sehr robuste Clusterzuweisungen. Die Samples aus Cluster 0 wurden identifiziert und sind in Tabelle 5.7 aufgelistet. Diese wurden als Ausreißer in Speichersatz 2 entfernt. Beim erneuten Iterieren bildeten sich 2 große und robuste Cluster. Deshalb wurde k schrittweise erhöht bis die Anzahl an Clustern einem Extrema der Indizes aus Abb. A.4b entspricht. Bei k=6 ergab sich ein weiteres kleines Cluster. Die Samples aus Cluster 4 wurden somit identifiziert (Tabelle 5.8) und in Speichersatz 3 entfernt. Für k=6 wurde in der Iteration zwar kein robustes Ergebnis erreicht, jedoch konnte für diesen Umstand (Tabelle A.1) eine recht kleine Anzahl an Springern identifiziert werden. Diese sind in Tabelle 5.9 zu finden. Nach dem Entfernen dieser, führte K-Means mit Speichersatz 4 wieder zu einem eindeutigen Ergebnis.

Tabelle 5.6: Clustergröße bei k=2

Speichersatz 1

	Loop	
cluster	0	21
0	6	5
1	395	396

Tabelle 5.7: Ausreißer aus Speichersatz 1

NUTS3	**Name**
DE212	München, Landeshauptstadt
DEB34	Ludwigshafen am Rhein, kreisfreie Stadt
DE600	Hamburg, Freie und Hansestadt
DE300	Berlin, Stadt
DEA23	Köln, Stadt
DE929	Region Hannover

Tabelle 5.8: Ausreiser aus Cluster 4 in Speichersatz 2 bei k=6

NUTS3	**Name**
DED21	Dresden, Stadt
DED51	Leipzig, Stadt
DE254	Nürnberg
DE712	Frankfurt am Main, Stadt

Tabelle 5.9: Identifizierte Springer in Speichersatz 3 bei k=6

NUTS3	**Name**
DE945	Wilhelmshaven, Stadt
DEA51	Bochum, Stadt
DEA52	Dortmund, Stadt
DE216	Bad Tölz-Wolfratshausen
DE125	Heidelberg, Stadtkreis
DE501	Bremen, Stadt
DEB35	Mainz, kreisfreie Stadt
DE244	Hof
DEG05	Weimar, Stadt

5.4 Ergebnisse

Aus den finalen Datensätzen und den Ergebnissen der Clusteranalyse werden mit der Methode der kleinsten Fehlerquadrate die Typregionen bestimmt. Die Regionen aus Tabelle 5.12 stehen exemplarisch für ihr Cluster, da sie mit ihren Werten am nächsten zum jeweiligen Clusterschwerpunkt sind. Die Medianwerte sind in Tabellen 5.10 und 5.11 aufgelistet. Vergleicht man die Clustergrößen der beiden Varianten fällt auf, dass die Cluster sehr ähnlich sind. 3 Cluster sind identisch, 2 Cluster weichen bei ihrerer Größe leicht voneinander ab und 1 Cluster hat andere Medianwerte und somit unterschiedliche Samples.

Tabelle 5.10: Medianwerte der finalen Clusterzuweisung von Variante 1

	Cluster					
	0	1	2	3	4	5
Einwohnerdichte [Pers/km2]	188.0	435.5	134.0	197.0	1646.0	1290.0
Fernwaermeabsatz GHD [GWh]	18.0	60.0	6.0	11.0	273.0	29.0
Jahresgasbedarf Industrie GWh	570.0	4343.0	171.0	164.0	309.0	162.0
Jahresgasbedarf Haushalte GWh	967.0	861.5	366.0	429.0	588.0	349.0
Siedlungsflaeche km2	134.0	156.0	59.0	70.0	61.0	30.0
Geothermiepotenzial	3.0	0.5	0.0	69.0	0.0	0.0
Clustergröße	83	10	157	65	21	45

Tabelle 5.11: Medianwerte der finalen Clusterzuweisung von Variante 2

	Cluster					
	0	1	2	3	4	5
Einwohnerdichte [Pers/km2]	184.5	435.5	134.0	197.0	1646.0	1290.0
Fernwaermeabsatz GHD [GWh]	18.0	60.0	6.0	11.0	287.0	29.0
Jahresgasbedarf Haushalte GWh	964.0	701.5	366.0	429.0	588.0	349.0
Jahresgasbedarf Industrie GWh	567.5	4725.5	171.0	164.0	309.0	162.0
Siedlungsflaeche km2	136.0	128.5	59.0	70.0	61.0	30.0
Geothermiepotenzial	3.5	0.5	0.0	69.0	0.0	0.0
Clustergröße	86	10	157	65	19	45

Bei genauerer Betrachtung lassen sich die Cluster typisieren. Cluster 3 steht repräsentativ für eine Gruppe mit hohem Geothermiepotenzial. Cluster 4 und 5 haben hohe Einwohnerdichten unterscheiden sich jedoch stark im Fernwärmeabsatz GHD. Cluster 0 hat den höchsten Jahresgasbedarf im Haushalt. Dicht gefolgt von Cluster 1, das aber durch einen extrem hohen Jahresgasbedarf in der Industrie heraussticht. Die größte Gruppe mit 157 Land- und Stadtkreisen befindet sich überall im Mittelfeld. Die Unterschiede der einzelnen Cluster pro

Feature sind in den Min-Max normierten Boxplots in Abb. A.8 und A.9 sehr gut zu erkennen. Diese Typen zeigen, dass die Clusteranalyse durchaus nachvollziehbar ist. Die Land- bzw. Stadtkreise diese Typen repräsentieren sind in Tabellen 5.12 und 5.13 aufgelistet. Nur für Cluster 1 unterscheiden sich diese.

Tabelle 5.12: Identifizierte Typregionen in Variante 1

cluster	NUTS3	Typregion	Clustergröße
0	DED2F	Sächsische Schweiz-Osterzgebirge	83
1	DEA1F	Wesel	10
2	DE127	Neckar-Odenwald-Kreis	157
3	DEB3B	Alzey-Worms	65
4	DE731	Kassel, documenta-Stadt	21
5	DE117	Heilbronn, Stadtkreis	45

Tabelle 5.13: Identifizierte Typregionen in Variante 2

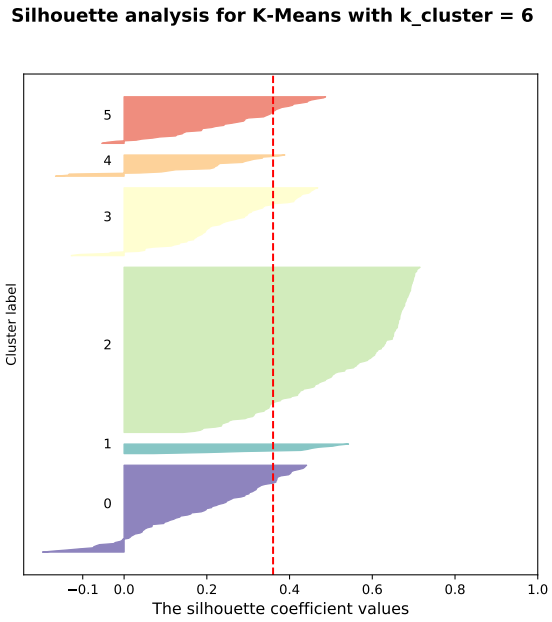
cluster	NUTS3	Typregion	Clustergröße
0	DED2F	Sächsische Schweiz-Osterzgebirge	86
1	DEC04	Saarlouis	10
2	DE127	Neckar-Odenwald-Kreis	157
3	DEB3B	Alzey-Worms	65
4	DE731	Kassel, documenta-Stadt	19
5	DE117	Heilbronn, Stadtkreis	45

Die Identifizierten Cluster werden in den regionalen Kontext gesetzt. Hierfür werden die Stadt- bzw. Landkreise ihrer Clusterzuordnung entsprechend, farblich differenziert in Abb. A.6a und A.6b als Karte dargestellt. Die aussortierte Samples werden in den Karten als weiße Flecken dargestellt. Diese haben keine eigene Clusterzuweisung durch K-Means erhalten, können aber als eigenständiges Cluster interpretiert werden. Auch die Typregionen werden in Abb. A.7a und A.7b abgebildet.

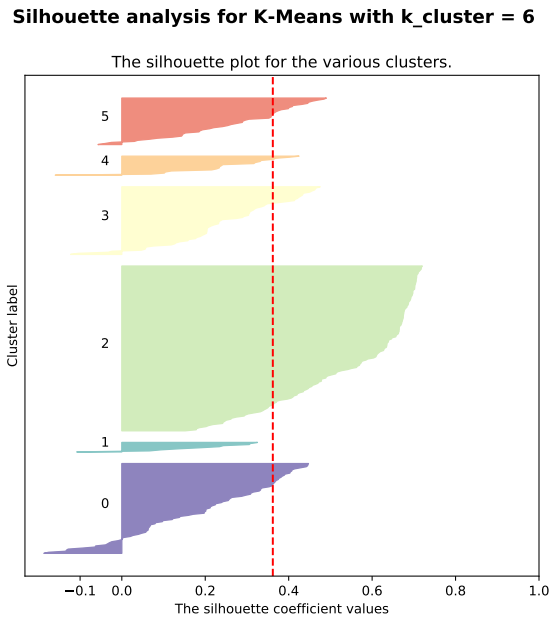
5.5 Diskussion

Insgesamt zeigt sich, dass das Modell in der Lage ist robuste Zuweisung zu tätigen. Es wurden zwei unterschiedlichen Verfahrensweisen durchgeführt, die beide ein fast identisches Ergebnis lieferten (Abb. A.6a und A.6b). Variante 2 bestätigt somit die Ausreißeridentifikation von Variante 1. Schritte wie die Feature-Selection und das Entfernen von Ausreißern können bei K-Means notwendig sein, wenn der Algorithmus zu einer eindeutigen Lösung kommen soll.

Besonders eine hohe Anzahl an Features stellt eine Herausforderung dar. Strukturen, in Datensätze mit deutlich höherer Samplezahl, sind hingegen einfacher zu identifizieren. Durch das Entfernen von Ausreißern und Clusterspringern reduziert sich die gültige Ergebnismenge. Ausreißer können jedoch als eigenes Cluster gewertet werden. Auch die Clusterspringer können nachträglich wieder zugeordnet werden. Obwohl K-Means in mehreren Versuchen eindeutig lösen konnte, bedingt dies nicht automatisch einer hohen Clustergüte. Anhand des Silhouettenplots kann die Güte für jedes Sample bestimmt werden.



(a) Silhouetten Plot mit für Variante 1



(b) Silhouetten Plot mit für Variante 2

Abbildung 5.1: Silhouetten Plots mit Silhouetten-Index für alle Samples pro Cluster

Quelle: eigene Darstellung

In Abb. 5.1a und 5.1b werden die beiden Varianten miteinander verglichen. Diese unterscheiden sich kaum voneinander. Negative Werte stehen für eine Falschzuordnung im nächsten Umfeld. Insgesamt fällt auf, dass ein Großteil des Samples unter dem Durchschnitt liegen. Dieser wird Maßgeblich von Cluster 2 beeinflusst und deutet auf eine zu geringe Clusteranzahl hin. Auch die starken Gradienten sprechen, nach diesem Maß, für eine geringe Homogenität im Cluster. Jedoch widersprachen die Inzides aus Abb. A.4a und A.4b dieser Vermutung. In der Zielfunktion von K-Means wird der euklidische Abstand zum Clusterschwerpunkt minimiert. Wie schon in Abschnitt 3.4 erwähnt, wird dadurch eine kugelförmigen Verteilung der Samples um den Clusterschwerpunkt vorausgesetzt. Liegt eine abweichende Verteilung vor, werden die Cluster unsauber voneinander getrennt.

6 Fazit und Ausblick

Ziel dieser Arbeit war es, eine Methode zu entwickeln, die es ermöglicht mit dem K-Means Clusteralgorithmus robuste Zuweisungen zu erlangen. Als partitionierendes Verfahren hat K-Means die Besonderheit, dass es in lokalen Minima terminieren kann. Hilfsmittel ermöglichen es, eine Datenauswahl zu treffen und diese so aufzubereiten, dass eine möglichst eindeutige Lösung gefunden werden kann. Hierbei stellt sich die Frage, ob das globale Minimum überhaupt das gewünschte Ziel ist. Mithilfe des Modells können unterschiedliche Varianten miteinander verglichen und bewertet werden. Außerdem wurden Funktionen integriert, die Teil- und Endergebnisse visualisieren. Die dafür notwendigen Methoden und ihre Implementierung beschrieben. Anhand einer exemplarischen Anwendung wurde das Verfahren noch einmal verdeutlicht. In beiden Varianten konnten robuste und eindeutige Lösungen gefunden werden. Anhand der Auswertung wurden abschließende die Grenzen des Modells aufgezeigt. Clusteralgorithmen werden meist bei großen Datensätzen verwendet. Der Verlust einiger Ausreißer kommt hierbei oft nicht ins Tragen. In dem Kontext der Arbeit lag die Herausforderung darin, bei einem sehr kleinen Datensatz, möglichst wenig Samples auszuschließen. Es sollten Typregionen identifiziert werden, die für ganz Deutschland stehen. Durch die Reduktion des Datensatzes auf eine Untermenge an gering korrelierenden Features, war es möglich, nur eine geringe Menge an Samples auszuschließen. Diese können als eigene Cluster gefasst oder nachträglich zugeordnet werden. Das Verwenden von weichen Clusterverfahren ist der Ausschluss von springenden Samples nicht erforderlich und würde somit die Ergebnismenge erweitern bzw. die nachträgliche Zuordnung vereinfachen. Um zu einem abschließenden Ergebnis bzgl. der Typregionen von Energieinfrastrukturen zu kommen, sind weitere Analysen notwendig. Durch den Vergleich von weiteren Feature-Untermengen können Erkenntnisse über den gesamten Ausgangsdatensatz erlangt werden. Sollten mehr Features eingebracht werden, sind dimensionsreduzierende Verfahren, wie die Faktorenanalyse oder Hauptkomponentenanalyse, von Vorteil. Diese wurden im Rahmen der Arbeit nicht betrachtet, stellen unter diesen Bedingungen jedoch einen erheblichen Mehrwert dar. Über die Güte andere Algorithmen ist an dieser Stelle keine Bewertung möglich. Der Ward-Algorithmus wird oft bei geringer Anzahl an Sample genutzt und würde auch hier passen. Ein dichtebasiertes-Verfahren würde das Modell, bei nicht kugelförmig verteilten Daten, ergänzen.

Literatur

4ÜNB (2019). *Netzentwicklungsplan Strom 2030*. URL: https://www.netzentwicklungsplan.de/de/netzentwicklungsplaene/netzentwicklungsplan-2030-2019 (besucht am 10. 12. 2019).

AEE (2019). *Endenergieverbrauch nach Strom, Wärme und Verkehr*. URL: https://www.unendlich-viel-energie.de/mediathek/grafiken/endenergieverbrauch-nach-strom-waerme-und-verkehr (besucht am 10. 12. 2019).

AGEB, Hrsg. (2017). *Zusammenfassung Anwendungsbilanzen für die Endenergiesektoren 2013 - 2017*. URL: https://ag-energiebilanzen.de/index.php?article_id=29&fileName=ageb_bericht_anwendungsbilanzen_2013-2017_final__2019-01-03.pdf (besucht am 18. 11. 2019).

Arbelaitz, O. u. a. (2013). „An extensive comparative study of cluster validity indices“. In: *Pattern Recognition* 46, S. 243–256.

Backhaus, K. u. a. (2016). „Clusteranalyse“. In: *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 453–516. ISBN: 978-3-662-46076-4. DOI: 10.1007/978-3-662-46076-4_9.

Bamberg, G., Baur, F. und Krapp, M. (2017). *Statistik-Arbeitsbuch: Übungsaufgaben - Fallstudien - Lösungen*. De Gruyter Studium. De Gruyter. ISBN: 9783110495751.

BDEW (2015). *Wie heizt Deutschland?* URL: https://www.bdew.de/media/documents/BDEW-Broschuere-Wie-heizt-Deutschland-2015.pdf (besucht am 10. 12. 2019).

Bellman, R. E. (2015). *Adaptive Control Processes: A Guided Tour*. Princeton Legacy Library. Princeton University Press. ISBN: 9781400874668.

BGR (2018). *Bericht zur Rohstoffsituation in Deutschland*. Bundesanstalt für Geowissenschaften und Rohstoffe. URL: https://www.bgr.bund.de/DE/Themen/Min_rohstoffe/Downloads/rohsit-2017.pdf (besucht am 10. 12. 2019).

BMVi (2019). *BMVi fördert technologieoffen CO2-freie Mobilität*. URL: https://www.unendlich-viel-energie.de/themen/akzeptanz-erneuerbarer/akzeptanz-umfrage/akzeptanzumfrage-2019 (besucht am 10. 12. 2019).

BMWi (2012). *Ausbau des Stromnetzes*. URL: https://www.bmwi.de/Redaktion/DE/Infografiken/Energie/abbildung-das-deutsche-stromnetz.html.

BMWi (2019a). *Rohöl: Transport, Lagerung und Verarbeitung.* URL: https://www.bmwi.de/Redaktion/DE/Artikel/Energie/mineraloel-transport-lagerung-verarbeitung.html (besucht am 10. 12. 2019).

BMWi, Hrsg. (2019b). *Versorgungssicherheit bei Erdgas. Monitoringbericht nach § 51 EnWG des Bundesministeriums für Wirtschaft und Energie.* URL: https://www.bmwi.de/Redaktion/DE/Publikationen/Energie/monitoringbericht-versorgungssicherheit-2017.pdf?__blob=publicationFile&v=24.

Bruns, E. u. a. (2012). *Netze als Rückgrat der Energiewende. Hemmnisse für die Integration erneuerbarer Energien in Strom-, Gas- und Wärmenetze.* DOI: http://dx.doi.org/10.14279/depositonce-3394.

Caliński, T. und Harabasz, J. (1974). „A dendrite method for cluster analysis". In: *Communications in Statistics* 3.1, S. 1–27. DOI: 10.1080/03610927408827101.

Chire (2017). *K-Means convergence.* URL: https://commons.wikimedia.org/wiki/File:K-means_convergence.gif (besucht am 10. 12. 2019).

comp.ai.neural-nets (2014). *Should I normalize, standardize, rescale the data.* URL: http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-16.html (besucht am 12. 12. 2019).

Davies, D. L. und Bouldin, D. W. (1979). „A Cluster Separation Measures". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2, S. 224–227. ISSN: 1939-3539. DOI: 10.1109/TPAMI.1979.4766909.

Destatis (2017). *Alle politisch selbständigen Gemeinden mit ausgewählten Merkmalen am 31.12.2017.* URL: https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/Archiv/GVAuszugJ/31122017_Auszug_GV.html (besucht am 10. 12. 2019).

eurostat (2018a). *NUTS - Systematik der Gebietseinheiten für die Statistik.* URL: https://ec.europa.eu/eurostat/de/web/nuts/background (besucht am 10. 12. 2019).

eurostat (2018b). *TERCET NUTS-postal codes.* NUTS 2016. URL: https://ec.europa.eu/eurostat/de/web/nuts/correspondence-tables/postcodes-and-nuts (besucht am 10. 12. 2019).

Geyler, S. u. a. (2008). *Schriftenreihe des Forschungsverbundes KoReMi.* Bd. 2: *Clusteranalyse der Gemeinden in der Kernregion Mitteldeutschland. Eine Typisierung der Region nach Entwicklungsparametern und Rahmenbedingungen.* Hrsg. von U. Leipzig.

Greenpeace (2019). *Wann Deutschland sein Klimaziel für 2020 tatsächlich erreicht.* Kurzgutachten. DIW. URL: https://www.greenpeace.de/sites/www.greenpeace.de/files/

publications/s02681_gp_energie_klimaziele_2020_studie_10_2019.pdf (besucht am 10. 12. 2019).

Halkidi, M., Batistakis, Y. und Vazirgiannis, M. (2001). „On Clustering Validation Techniques“. In: *Journal of Intelligent Information Systems* 17.2, S. 107–145. ISSN: 1573-7675. DOI: 10.1023/A:1012801612483.

Hastie, T., Tibshirani, R. und Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer. ISBN: 9780387848846.

Herink, M. und Petersen, V. (2004). „Clusteranalyse als Instrument zur Gruppierung von spezialisierten Marktfruchtunternehmen“. In: *Agrarwirtschaft* 53.7. Hrsg. von D. Fachverlage. URL: http://www.gjae-online.de/news/pdfstamps/freeoutputs/GJAE-326_2004.pdf (besucht am 10. 12. 2019).

IWES und IBP (2017). *Wärmewende 2030. Schlüsseltechnologien zur Erreichung der mittel- und langfristigen Klimaschutzziele im Gebäudesektor. Studie im Auftrag von Agora Energiewende*. URL: https://www.agora-energiewende.de/fileadmin2/Projekte/2016/Sektoruebergreifende_EW/Waermewende-2030_WEB.pdf.

Jäckle, S. (2017). *Neue Trends in den Sozialwissenschaften: Innovative Techniken für qualitative und quantitative Forschung*. Springer Fachmedien Wiesbaden. ISBN: 9783658171896. URL: https://books.google.de/books?id=Sla-DgAAQBAJ (besucht am 10. 12. 2019).

Keles, D. u. a. (2017). „Meeting the Modeling Needs of Future Energy Systems“. In: *Energy Technology* 5.7, S. 1007–1025. DOI: 10.1002/ente.201600607.

Kendall, M. G. (1970). *Rank Correlation Methods*. 4. Aufl. Griffin, London.

Lilliefors, H. W. (1967). „On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown“. In: *Journal of the American Statistical Association* 62.318, S. 399–402. DOI: 10.1080/01621459.1967.10482916.

Linsenmeier, M. (2017). „Estimating cost of extending electricity distribution networks in Germany“. Magisterarb. School of Business und Economics of Humboldt-Universität zu Berlin.

Liu, Y. u. a. (2010). „Understanding of Internal Clustering Validation Measures“. In: *2010 IEEE International Conference on Data Mining*. IEEE. DOI: 10.1109/icdm.2010.35.

Lloyd, S. (1982). „Least Squares Quantization in PCM“. In: *IEEE Trans. Inf. Theor.* 28.2, S. 129–137. ISSN: 0018-9448. DOI: 10.1109/TIT.1982.1056489.

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press. ISBN: 9780521642989.

MacQueen, J. (1967). „Some methods for classification and analysis of multivariate observations“. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Statistics*. Bd. 1. Berkeley, Calif.: University of California Press, S. 281–297. URL: https://projecteuclid.org/euclid.bsmsp/1200512992.

Milligan, G. W. und Cooper, M. C. (1985). „An examination of procedures for determining the number of clusters in a data set“. In: *Psychometrika* 50.2, S. 159–179. ISSN: 1860-0980. DOI: 10.1007/BF02294245.

Raghav, R. V., Lemaitre, G. und Unterthiner, T. (2019). *Compare the effect of different scalers on data with outliers*. URL: https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html (besucht am 10. 12. 2019).

Robinius, M., Markewitz, P. und Lopion, P. (2019). *Kosteneffiziente und klimagerechte Transformationsstrategien für das deutsche Energiesystem bis zum Jahr 2050. (Kurzfassung)*. Hrsg. von F. J. GmbH. URL: https://fz-juelich.de/iek/iek-3/DE/_Documents/Downloads/transformationStrategies2050_studySummary_2019-10-31.pdf.pdf;jsessionid=2A0309D0FE68226B2932FF02EEB67512?__blob=publicationFile.

Rousseeuw, P. J. (1987). „Silhouettes: A graphical aid to the interpretation and validation of cluster analysis“. In: *Journal of Computational and Applied Mathematics* 20, S. 53–65. ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-0427(87)90125-7.

Sambandam, R. (2013). *Marketing Research*. Bd. 15: *Cluster Analysis Gets Complicated*. Hrsg. von A. M. Association. Kap. 1. URL: https://www.trchome.com/docs/5-cluster-analysis-gets-complicated/file (besucht am 10. 12. 2019).

Stein, P. und Vollnahls, S. (2012). *Grundlagen clusteranalytischer Verfahren*. URL: https://www.uni-due.de/imperia/md/content/soziologie/stein/skript_clusteranalyse_sose2011.pdf (besucht am 10. 12. 2019).

UBA (2017). *Endenergieverbrauch 2017 nach Sektoren und Energieträgern*. Umweltbundesamt auf Basis Arbeitsgemeinschaft Energiebilanzen. URL: https://www.umweltbundesamt.de/daten/energie/energieverbrauch-nach-energietraegern-sektoren.

Vendramin, L., Campello, R. J. G. B. und Hruschka, E. R. (2010). „Relative clustering validity criteria: A comparative overview“. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 3.4, S. 209–235. DOI: 10.1002/sam.10080.

Wall, W. (2016). „Energetisch vergleichbare Städtegruppe. eine gesamtheitliche Clusteranalyse und Clusterauswahl deutscher kreisfreier Städte auf Basis der typischen Verbrauchssektoren und sozio-energetischer Indikatoren“. Diss. Ruhr-Universität Bochum, Lehrstuhl Energiesysteme und Energiewirtschaft. ISBN: 978-3-934951-41-9.

Weinand, J. M., McKenna, R. und Fichtner, W. (2019). „Developing a municipality typology for modelling decentralised energy systems“. In: *Utilities Policy* 57, S. 75–96. ISSN: 0957-1787. DOI: https://doi.org/10.1016/j.jup.2019.02.003.

Wiedenbeck, M. und Züll, C. (2010). „Clusteranalyse“. In: *Handbuch der sozialwissenschaftlichen Datenanalyse*. Hrsg. von C. Wolf und H. Best. VS Verlag für Sozialwissenschaften, S. 525–552. ISBN: 978-3-531-92038-2. DOI: 10.1007/978-3-531-92038-2_21.

Zimek, A., Schubert, E. und Kriegel, H.-P. (2012). „A survey on unsupervised outlier detection in high-dimensional numerical data“. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 5.5, S. 363–387. DOI: 10.1002/sam.11161.

A Anhang A

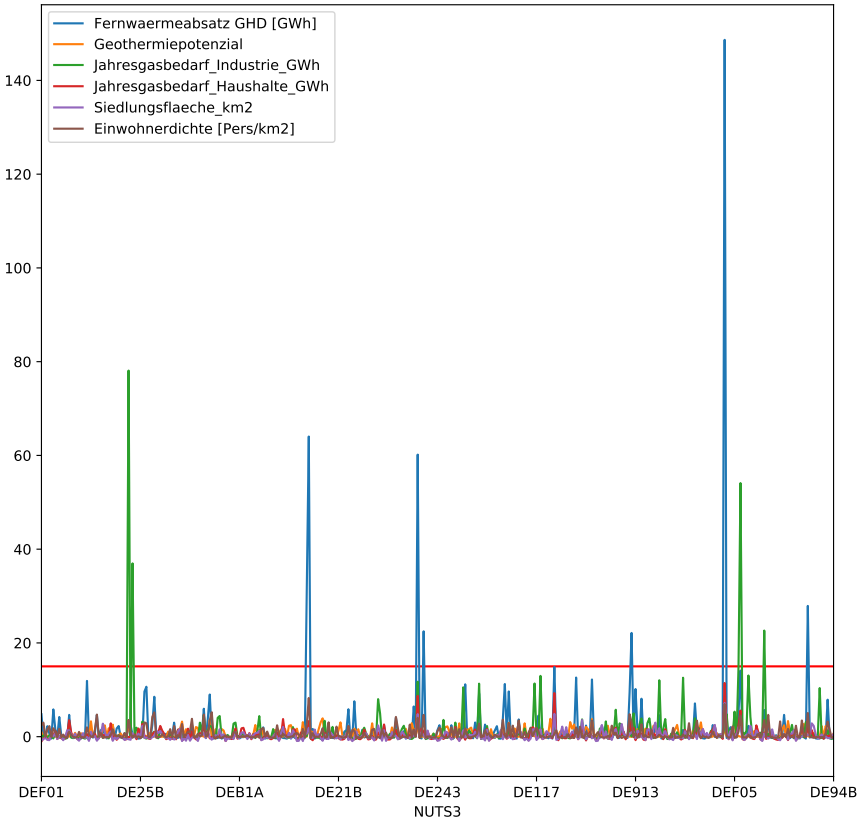
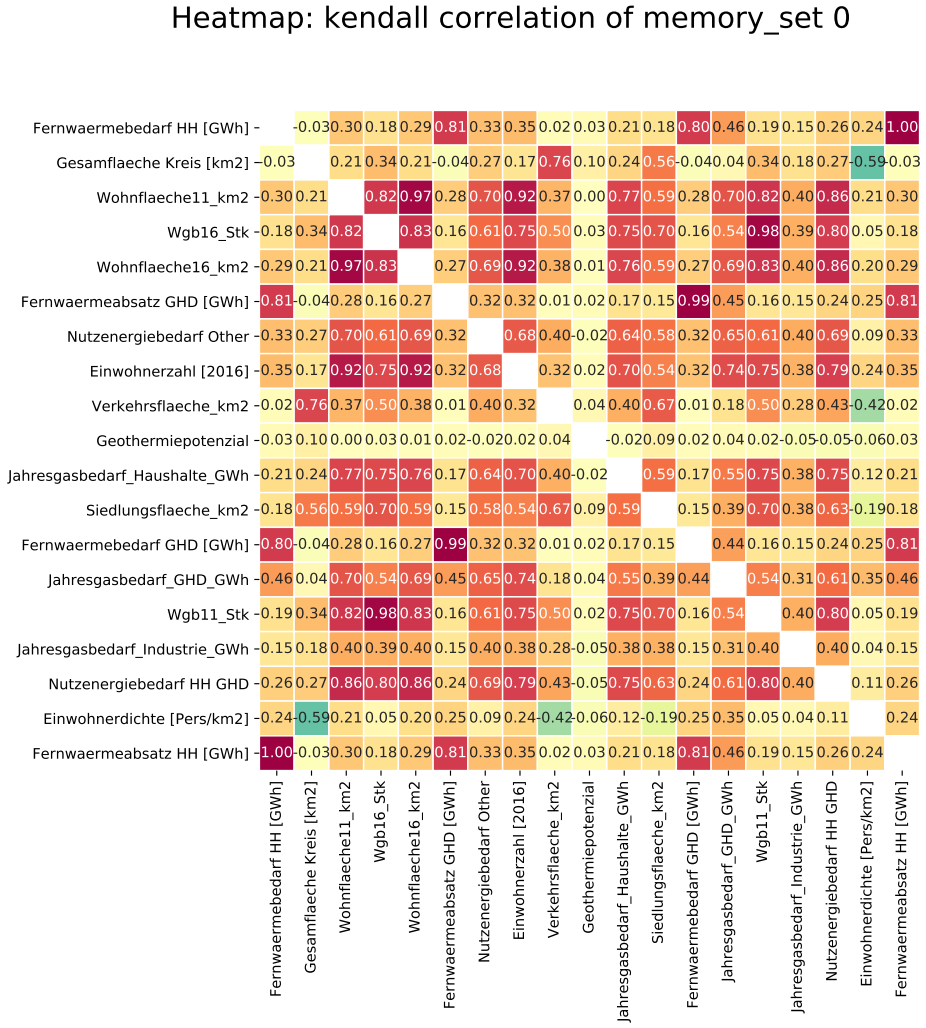
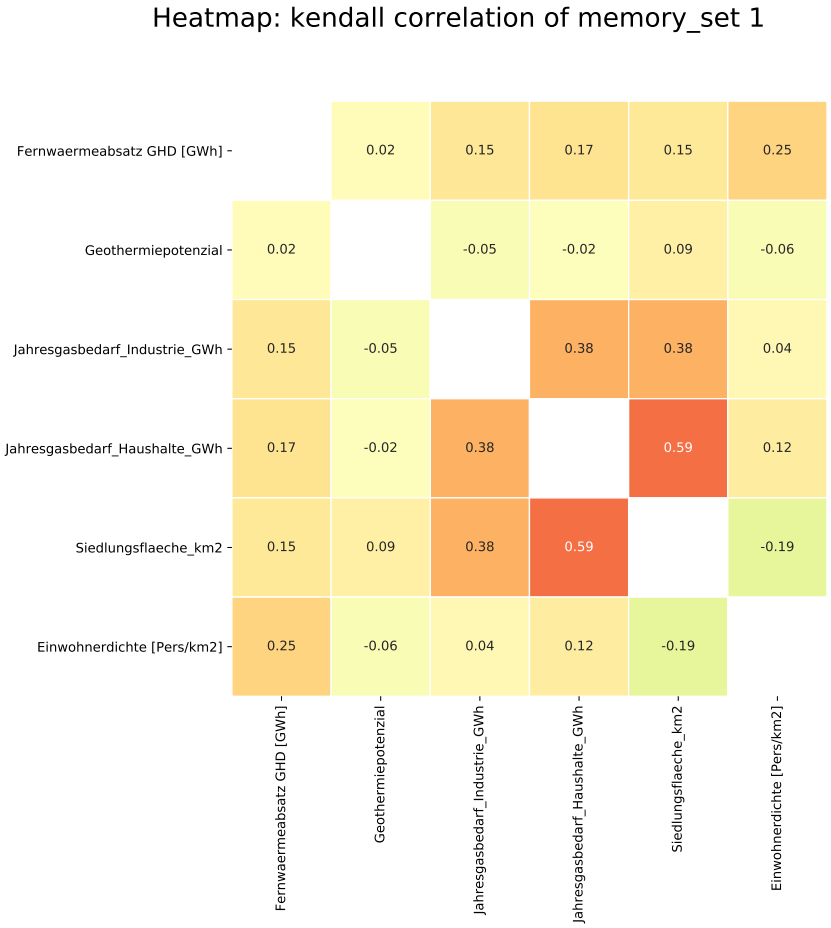


Abbildung A.1: Bestimmung der Ausreißer in Variante 1 mit Grenzwert=15

Quelle: eigene Darstellung



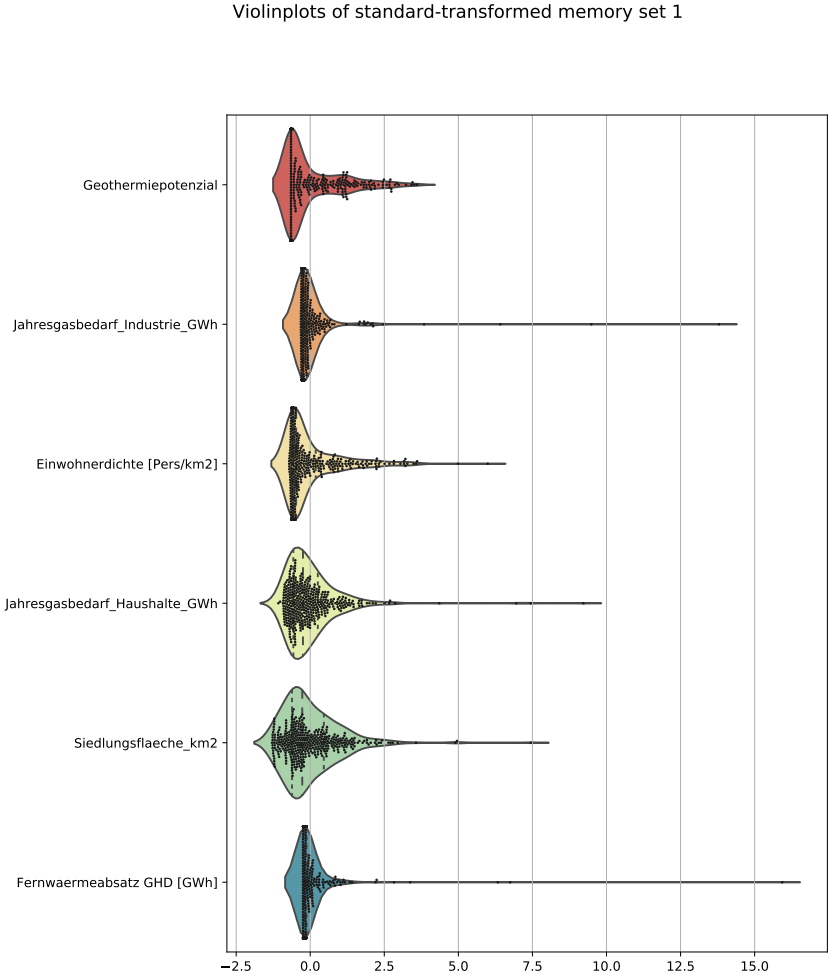
(a) Korrelationswerte des gesamten Datensatzes



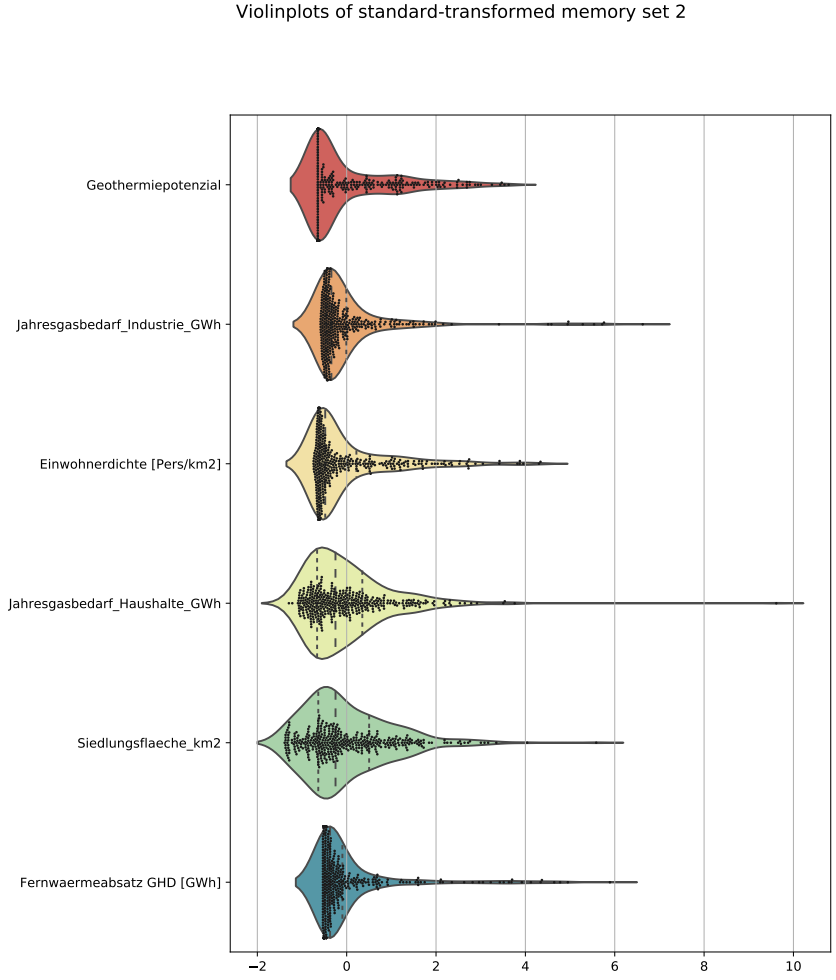
(b) Korrelationsanalyse des ausgewählten Datensatzes

Abbildung A.2: Korrelationsanalysen nach Kendall

Quelle: eigene Darstellung



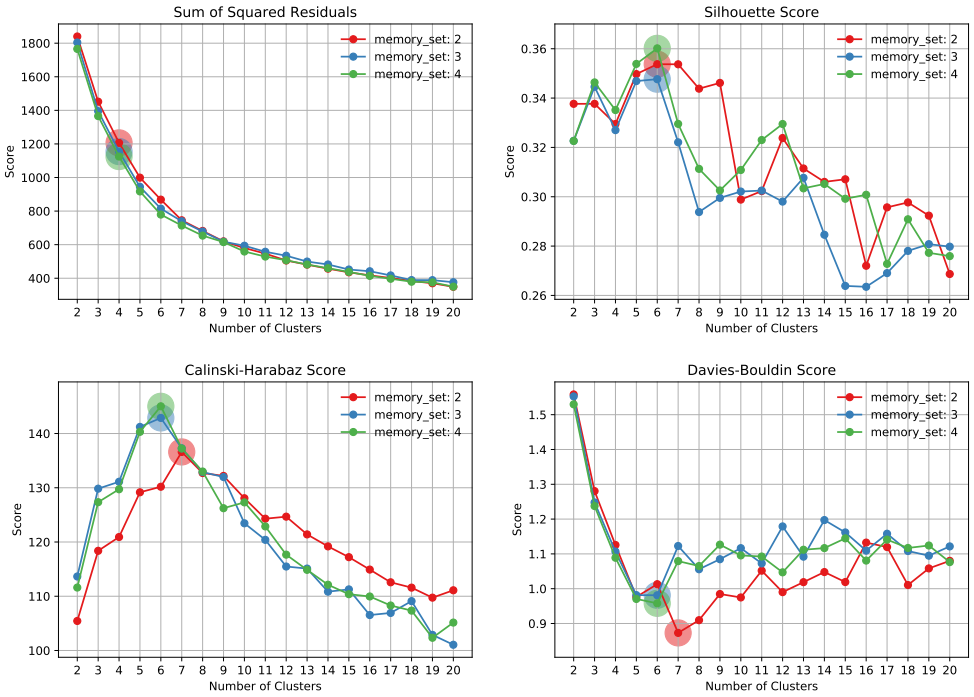
(a) Daten vor dem Entfernen



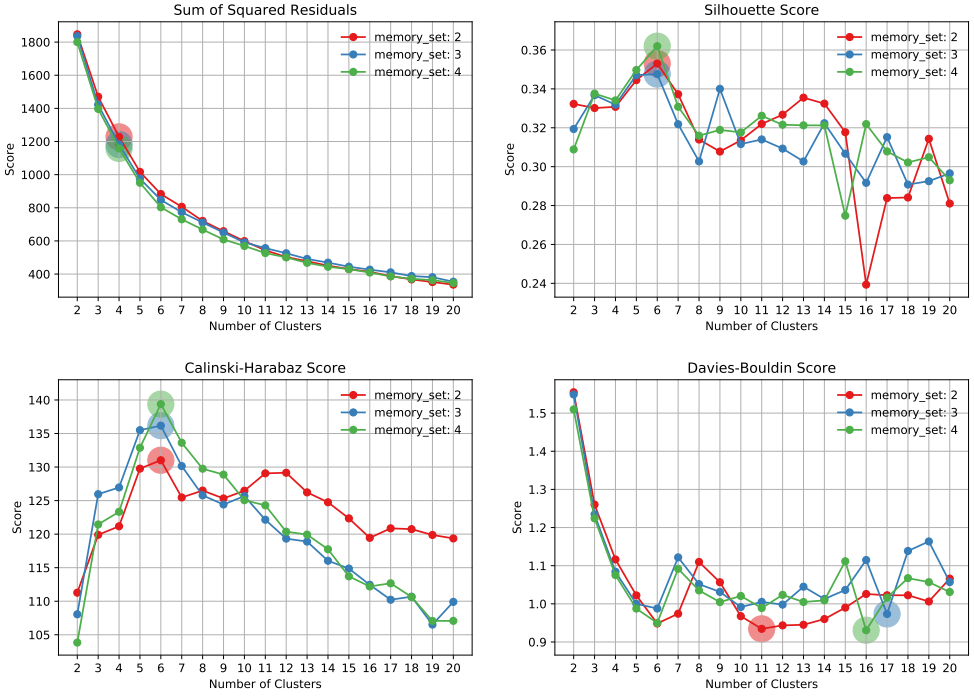
(b) Daten vor dem Entfernen

Abbildung A.3: Standardisierte Daten vor und nach der Ausreißeridentifikation in Variante 1

Quelle: eigene Darstellung



(a) Validierungsindizes für die Clusteranzahl k 2-20 der Speichersätze 2,3 und 4 in Variante 1



(b) Validierungsindizes für die Clusteranzahl k 2-20 der Speichersätze 2,3 und 4 in Variante 2

Quelle: eigene Darstellung

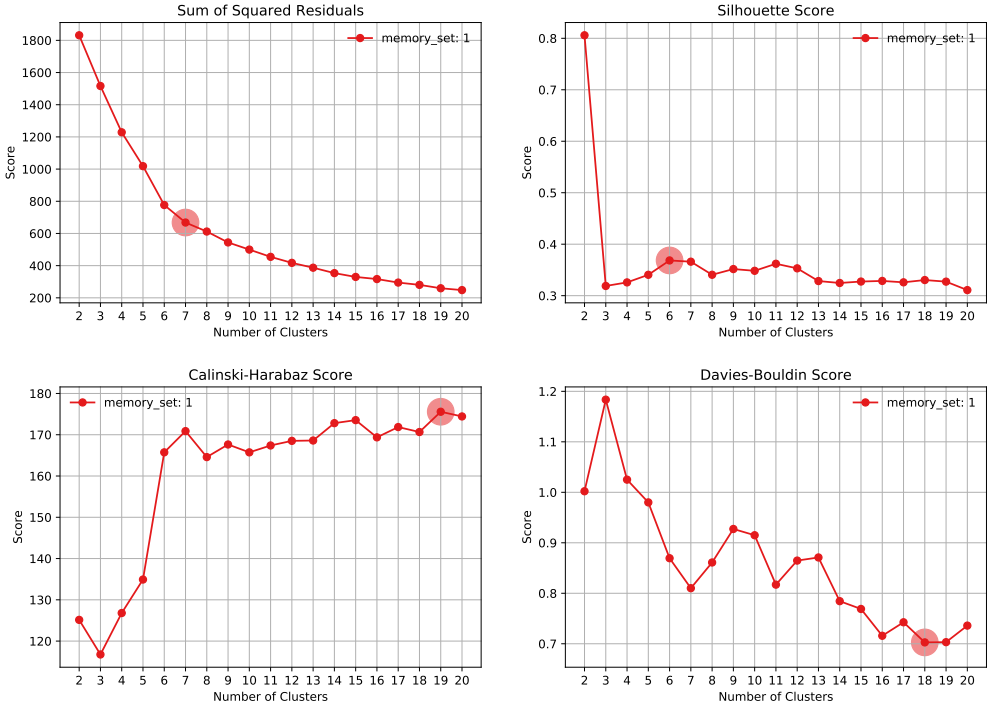
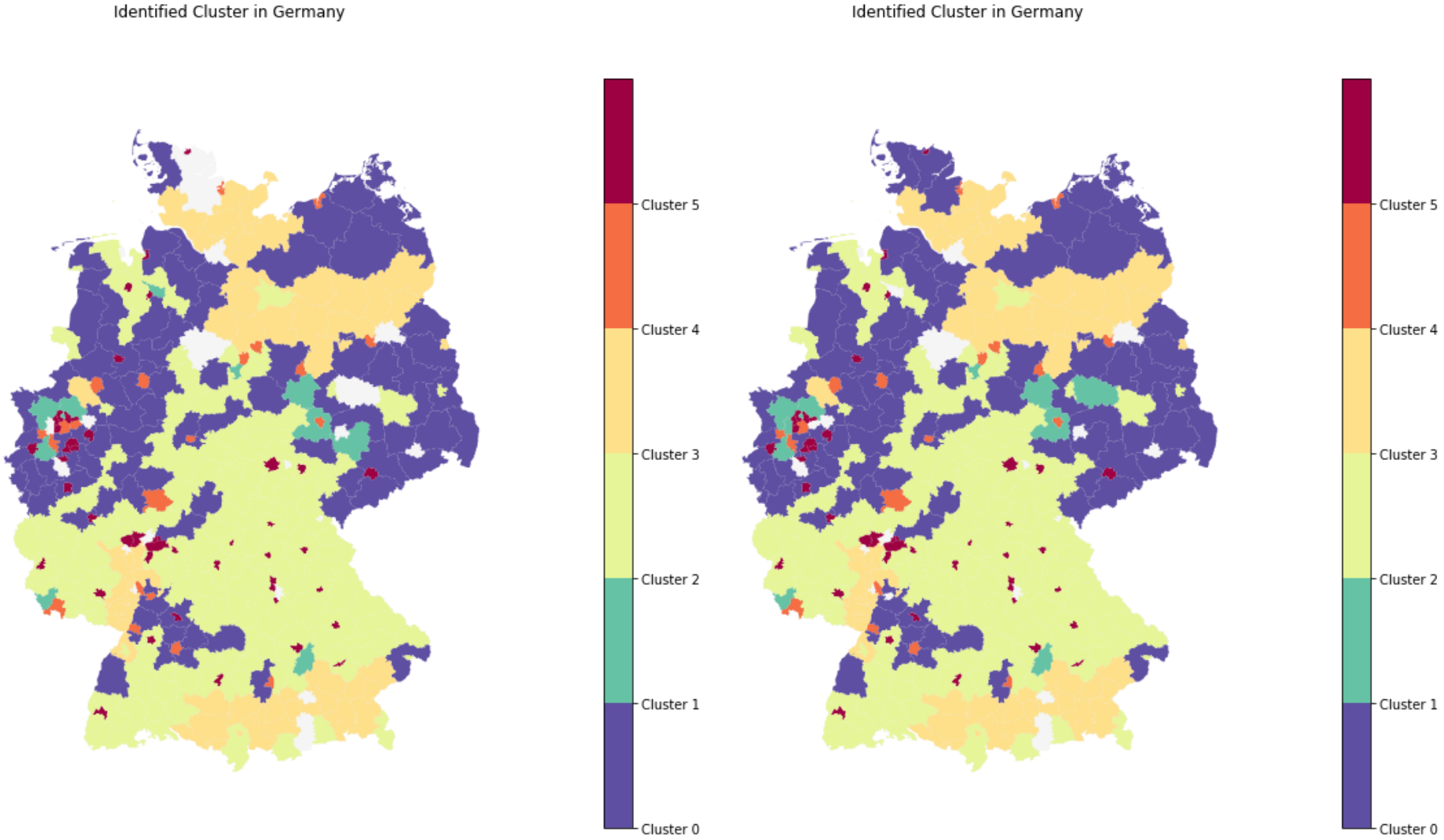


Abbildung A.5: Validierungsindizes für die Clusteranzahl k 2-20 von Speichersatz 1 in Variante 2

Quelle: eigene Darstellung

Tabelle A.1: Clustergrößen der Iteration von Speichersatz 4 bei k=6

	Loop														
cluster	0	1	2	3	4	5	6	7	9	10	15	21	22	23	26
0	67	66	65	66	66	66	67	67	65	66	67	65	66	65	66
1	22	22	23	24	23	23	23	22	20	22	22	22	22	22	23
2	159	160	161	158	158	159	157	157	161	160	158	161	159	161	160
3	10	10	10	10	11	10	11	11	11	11	10	10	10	11	10
4	47	47	46	47	47	47	47	48	47	46	48	47	48	46	46
5	86	86	86	86	86	86	86	86	87	86	86	86	86	86	86

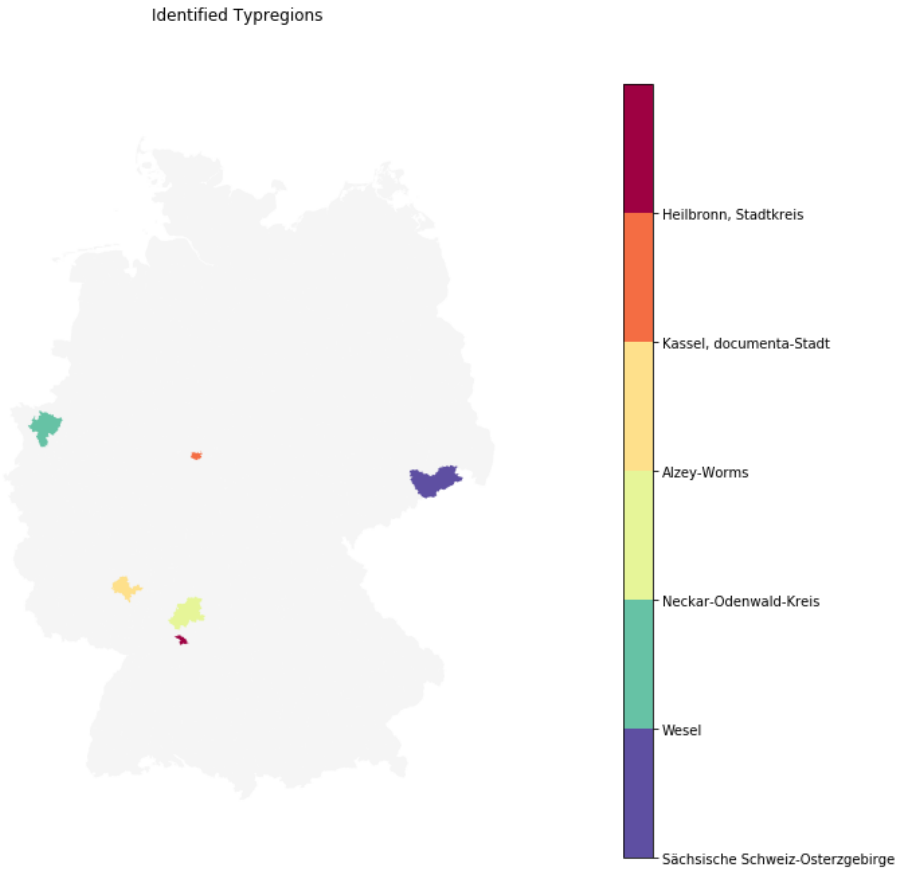


(a) Identifizierte Cluster in Variante 1

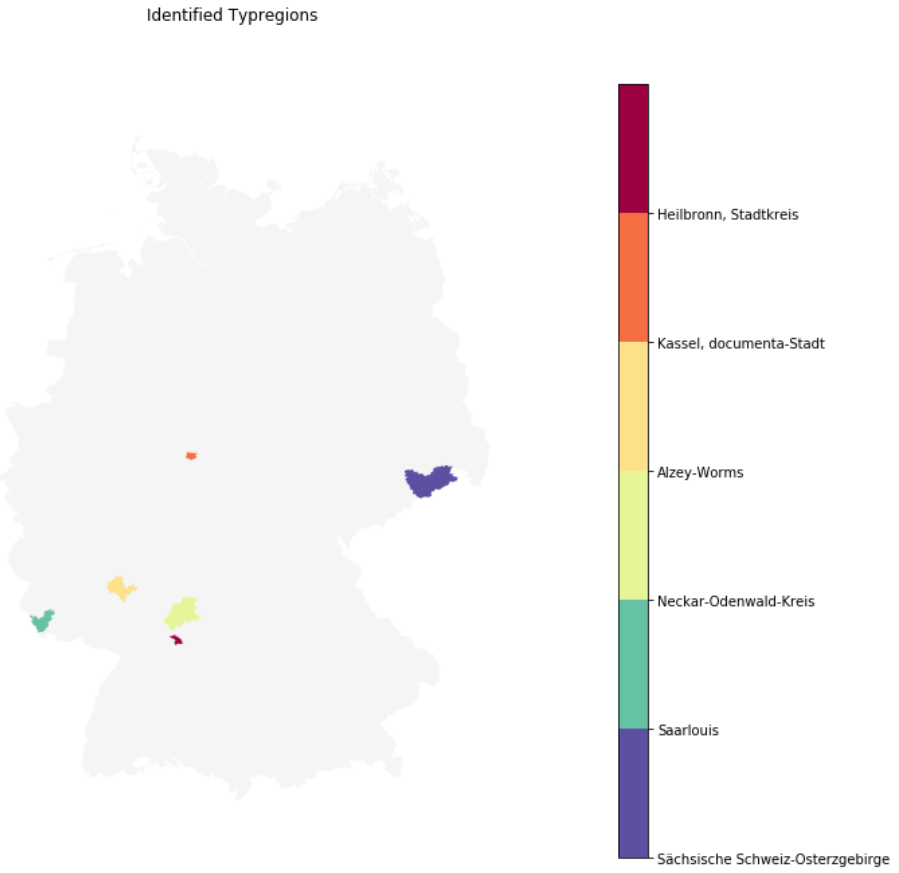
(b) Identifizierte Cluster in Variante 2

Abbildung A.6: Clusterkarten der beiden Varianten

Quelle: eigene Darstellung



(a) Identifizierte Typregionen in Variante 1



(b) Identifizierte Typregionen in Variante 2

Abbildung A.7: Typregionen der beiden Varianten

Quelle: eigene Darstellung

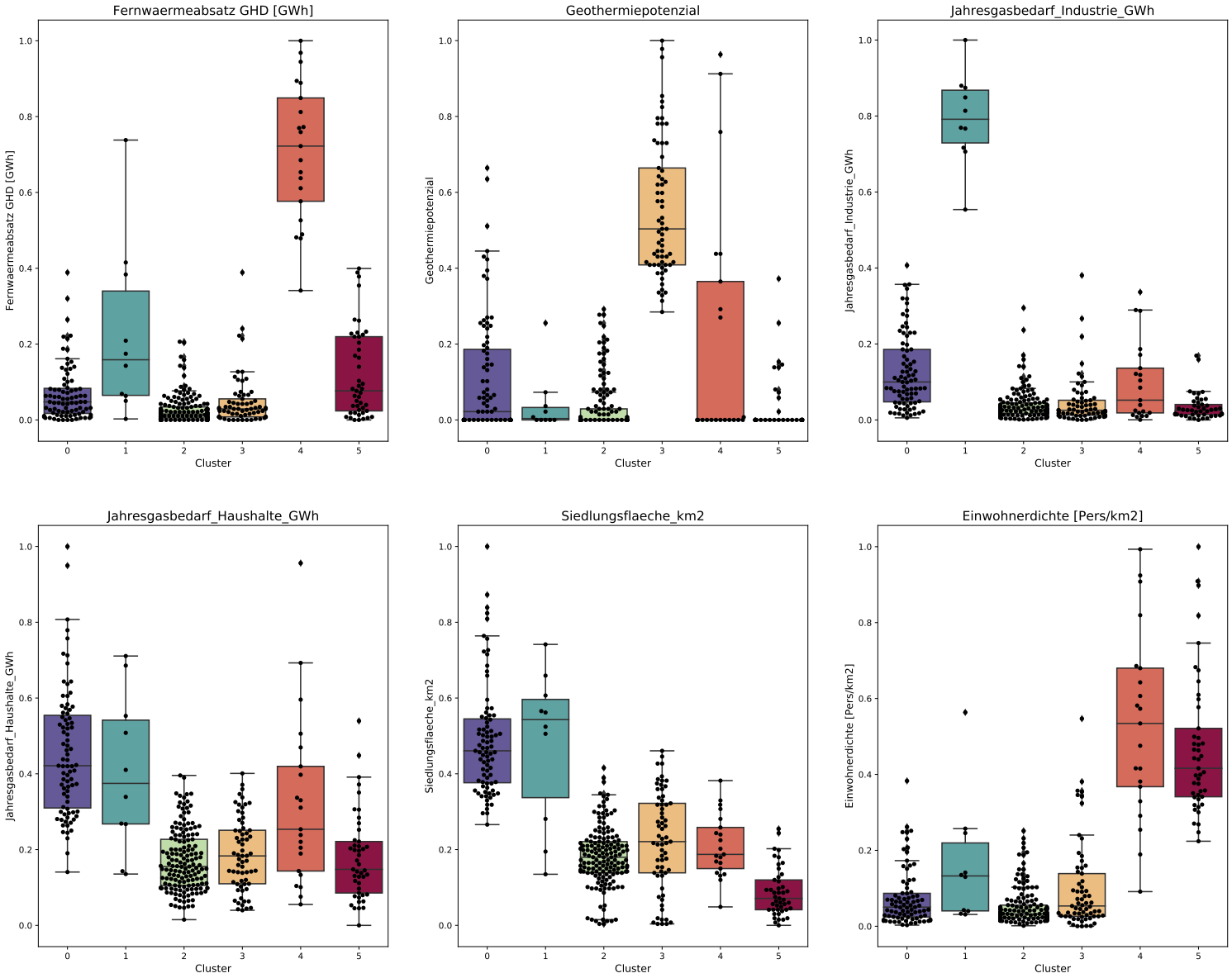


Abbildung A.8: Darstellung der unterschiedlichen Cluster von Variante 1 pro Feature

Quelle: eigene Darstellung

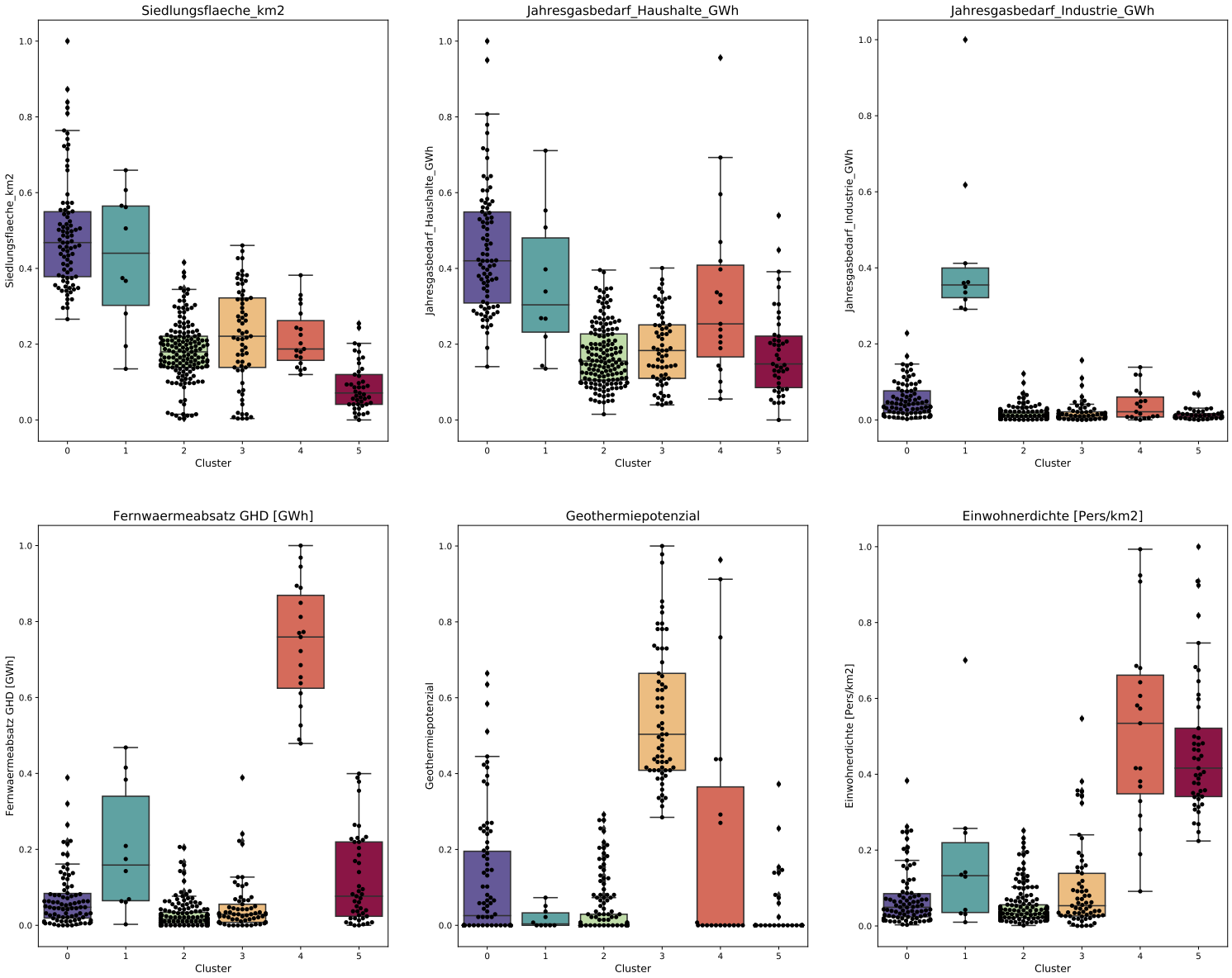


Abbildung A.9: Darstellung der unterschiedlichen Cluster von Variante 2 pro Feature

Quelle: eigene Darstellung