Vergleich von häufigsten Clustering-Algorithmen auf Kaggle

Florian Merlau Friedrich-Alexander Universität
Erlangen, Deutschland
Email: florian.merlau@fau.de

***Zusammenfassung*—Diese Arbeit präsentiert eine systematische Untersuchung zum Einsatz und zur Performance gängiger Clustering-Algorithmen im Kontext von Kaggle-Wettbewerben. Dazu wurden aus öffentlich verfügbaren Kaggle-Notebooks Datensätze und Implementierungen extrahiert, die auf unüberwachtes Lernen abzielen. Im Fokus standen partitionierende, hierarchische sowie probabilistische Verfahren, darunter K-Means, Agglomerative Clustering, Spectral Clustering, Gaussian Mixture Models und Bayesian Gaussian Mixture Models. Über verschiedene Wettbewerbe hinweg zeigt sich, dass K-Means aufgrund seiner leichten Implementierbarkeit und geringen Rechenkosten am häufigsten zum Einsatz kommt, während probabilistische Modelle insbesondere bei komplexen Datenstrukturen und Ausreißern robuste Ergebnisse liefern. Hierarchische und spektrale Ansätze punkten bei nicht-konvexen oder hochdimensionalen Datensätzen, erfordern jedoch umfangreichere Rechen- und Abstimmungsaufwände. Insgesamt verdeutlichen die Ergebnisse, dass die Auswahl des optimalen Clustering-Algorithmus entscheidend von den jeweiligen Dateneigenschaften und Anforderungen abhängig ist.**

***Index Terms*—Clustering, Unüberwachtes Lernen, Data Science, Kaggle, K-Means, Spectral Clustering, Agglomerative Clustering, Gaussian Mixture Models, Bayesian Gaussian Mixture Models**

I. Einführung

Die stetig wachsende Verfügbarkeit großer und komplexer Datensätze stellt sowohl Forschung als auch Industrie vor neue Herausforderungen. Eine zentrale Aufgabe in diesem Kontext ist es, verborgene Strukturen in den Daten zu identifizieren und diese in sinnvolle, homogene Gruppen zu unterteilen. Diese Methode, die als Clusteranalyse bezeichnet wird, ermöglicht es, Muster und Zusammenhänge zu erkennen, ohne dass vorab definierte Kategorien vorliegen. [1]

In der vorliegenden Arbeit wird ein allgemeiner Überblick über den Einsatz verschiedener Clustering-Algorithmen gegeben, wie sie in einem praxisnahen Wettbewerbsumfeld zum Einsatz kommen. Anhand öffentlich zugänglicher Daten werden exemplarisch verschiedene Verfahren betrachtet, ohne dabei zu sehr in technische Details zu gehen. Ziel ist es, aufzuzeigen, welche Algorithmen in einem realen Szenario häufig verwendet werden und inwieweit ihre Leistungsfähigkeit den Anforderungen moderner Datenanalysen gerecht wird.

Die Untersuchung stützt sich auf eine Analyse, die einen breiten Rahmen der Thematik abdeckt. Dabei werden die grundlegenden Konzepte der Clusteranalyse erläutert und die angewandten Methoden in einem wettbewerbsorientierten Umfeld diskutiert. Die Arbeit verzichtet bewusst darauf, sich zu sehr in spezifische technische Details einzelner Algorithmen zu verlieren, um einen allgemeinen und verständlichen Überblick zu ermöglichen.

Der Aufbau dieser Arbeit gliedert sich wie folgt: In Abschnitt II werden zunächst verwandte Arbeiten und bestehende Ansätze im Bereich der Clusteranalyse vorgestellt. Anschließend vermittelt Abschnitt III die theoretischen Grundlagen sowie zentrale Konzepte gängiger Clustering-Verfahren. In Abschnitt IV wird die methodische Vorgehensweise dieser Untersuchung detailliert beschrieben, bevor in Abschnitt V die praktische Durchführung der Analyse erläutert wird. Im weiteren Verlauf werden in den Kapiteln zu den spezifischen Clustering-Algorithmen und deren Vergleich (siehe *Clustering-Algorithmen* und *Vergleich der Clustering-Algorithmen – Eine differenzierte Betrachtung*) die Ergebnisse im Detail präsentiert. Abschließend fassen Abschnitt VIII die wesentlichen Erkenntnisse zusammen, und in Abschnitt IX wird ein Ausblick auf zukünftige Forschungsansätze gegeben.

II. Verwandte Arbeiten

Die Clusteranalyse und ihre Anwendungen im Bereich des maschinellen Lernens und der Datenanalyse stellen seit Jahrzehnten ein zentrales Forschungsgebiet dar. Ein klassischer Überblick über die verschiedenen Ansätze – von partitionierenden und hierarchischen bis hin zu dichtebasierten und modellbasierten Verfahren – wird in [2] präsentiert, wo eine umfassende Taxonomie der Methoden vorgestellt wird. Xu und Wunsch [1] erweitern diese Perspektive, indem sie insbesondere die Herausforderungen bei der Skalierbarkeit und der Robustheit von Clustering-Algorithmen in hochdimensionalen Datenräumen diskutieren.

Ein weiterer wesentlicher Beitrag in diesem Kontext stammt von Ng et al. [3], die in ihrer Arbeit nicht nur die theoretischen Grundlagen des Spectral Clustering beleuchten, sondern auch praktische Implementierungsaspekte und Anwendungsszenarien – etwa in der Bildsegmentierung – aufzeigen. Für hierarchische Clustering-Methoden liefern Kaufman und Rousseeuw [4] fundierte Ergebnisse, die sowohl die theoretische Herleitung als auch die Evaluierung von agglomerativen Verfahren umfassen.

Im Bereich probabilistischer Modelle wird das Gaussian Mixture Model (GMM) als leistungsfähiger Ansatz zur Modellierung von Daten als Mischung von Gaußverteilungen

Dieser Beitrag entstand im Rahmen des “Big Data Seminar”s, das im Wintersemester 2024/2025 vom Lehrstuhl für Informatik 6 (Datenmanagement) der Friedrich-Alexander Universität Erlangen-Nürnberg durchgeführt wurde.

etabliert [5]. Die Erweiterung dieses Konzepts durch bayesianische Inferenz, die in Bayesian Gaussian Mixture Models (BGMM) resultiert, wird in [6] behandelt. Diese Methoden bieten den Vorteil, Unsicherheiten in der Modellierung explizit zu berücksichtigen und sind besonders nützlich, wenn die Anzahl der Cluster im Vorfeld nicht klar definiert ist.

III. Grundlagen

Clustering ist eine Methode, um Daten in Gruppen zu unterteilen, in denen die Datenpunkte innerhalb einer Gruppe einander ähnlich sind. Dabei werden keine vorgegebenen Kategorien genutzt, sondern die Gruppen (Cluster) werden durch das Muster in den Daten selbst bestimmt [1]. Diese Methode hilft dabei, Zusammenhänge und Strukturen in großen Datensammlungen zu entdecken.

Grundsätzlich kann man Clustering-Verfahren in verschiedene Kategorien einteilen:

A. Partitionierende Verfahren

Bei diesen Verfahren wird der Datensatz in eine festgelegte Anzahl von Gruppen aufgeteilt. Man kann sich das vorstellen wie das Aufteilen eines Haufens von Gegenständen in genau definierte Schubladen, wobei jeder Gegenstand in die Schublade kommt, die ihm am ähnlichsten ist [2].

B. Hierarchische Verfahren

Hier wird eine Art Baumstruktur erstellt, in der zuerst kleine Gruppen gebildet und diese anschließend zu größeren Gruppen zusammengefasst werden – ähnlich wie bei einem Stammbaum, in dem Verwandte in engeren und weiteren Beziehungen dargestellt werden [4].

C. Dichtebasierte Methoden

Diese Methoden gruppieren Datenpunkte, die in ihrer Umgebung sehr dicht beieinander liegen. Das heißt, Punkte, die nahe zueinander liegen, werden zu einem Cluster zusammengefasst, während einzelne, weit entfernte Punkte als Ausreißer erkannt werden können [7].

D. Weitere Ansätze

Neben den oben genannten Kategorien gibt es auch Ansätze, die davon ausgehen, dass die Daten aus verschiedenen „Quellen“ oder Wahrscheinlichkeitsverteilungen stammen. Außerdem gibt es Methoden, bei denen ein Datenpunkt gleichzeitig mehreren Gruppen zugeordnet werden kann, was besonders bei unscharfen Übergängen nützlich ist [8].

E. Bewertung der Clustering-Ergebnisse

Die Qualität der Clusterbildung kann mit verschiedenen Metriken bewertet werden. Zwei gängige Methoden sind der Silhouettenkoeffizient und der Adjusted Rand Index (ARI).

a) Silhouettenkoeffizient:: Diese Metrik misst, wie ähnlich ein Datenpunkt zu den Punkten in seinem eigenen Cluster im Vergleich zu denen in anderen Clustern ist. [9].

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

b) Adjusted Rand Index (ARI):: Der ARI bewertet die Übereinstimmung zwischen der durch das Clustering erhaltenen Gruppierung und einer bekannten Referenzklassifikation. Er berücksichtigt dabei die Anzahl der Punktpaare, die in beiden Gruppierungen gleich behandelt werden (zusammen oder getrennt). [10]

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}},$$

F. Kaggle

Kaggle ist eine Online-Community-Plattform für Datenwissenschaftler und Machine-Learning-Enthusiasten. Sie ermöglicht es Nutzern, miteinander zu kollaborieren, Datensätze zu finden und zu veröffentlichen, GPU-integrierte Notebooks zu verwenden und an Wettbewerben teilzunehmen, um datenwissenschaftliche Herausforderungen zu lösen. [11]

Ein zentrales Element von Kaggle sind die Wettbewerbe, bei denen Unternehmen und Organisationen große Mengen an Daten und herausfordernde Aufgabenstellungen bereitstellen. Teilnehmer konkurrieren dabei, um die besten Modelle zur Lösung dieser Aufgaben zu entwickeln. Diese Wettbewerbe bieten nicht nur die Möglichkeit, praktische Erfahrungen zu sammeln, sondern auch, von anderen zu lernen und innovative Ansätze zu entdecken [12].

IV. Methodik

Dieser Abschnitt beschreibt das methodische Vorgehen zur Identifizierung und Analyse von Kaggle-Wettbewerben mit Fokus auf Clustering-Verfahren. Ziel ist es, die Häufigkeit des Einsatzes verschiedener Clustering-Algorithmen zu ermitteln und deren relative Leistungsfähigkeit zu bewerten.

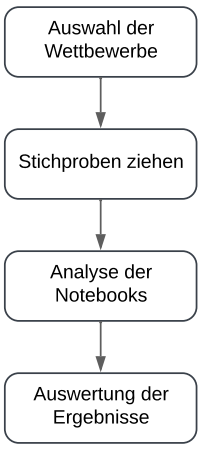

Abbildung 1. Ablaufdiagramm der Methodik (eigene Darstellung)

A. Fokus auf Clustering-Wettbewerbe

Der Schwerpunkt dieser Untersuchung liegt auf Wettbewerben, die speziell für Clustering-Aufgaben konzipiert sind. Clustering ist eine Methode des unüberwachten Lernens, bei

der ähnliche Datenpunkte ohne vorherige Labels in Gruppen eingeteilt werden (vgl. Kapitel III). Durch die Fokussierung auf derartige Wettbewerbe wird sichergestellt, dass die analysierten Notebooks relevante Clustering-Algorithmen implementieren. Dadurch wird vermieden, Datensätze einzubeziehen, die für andere Analyseformen vorgesehen sind, was aussagekräftige Ergebnisse im Bereich des unüberwachten Lernens garantiert.

B. Auswahl der Wettbewerbe

Die Identifikation relevanter Wettbewerbe erfolgt über die Filterfunktion der Kaggle-Plattform. Mithilfe dieser Funktion werden gezielt Wettbewerbe herausgefiltert, die für Clustering-Aufgaben geeignet sind. Dabei werden folgende Auswahlkriterien angewendet:

- **Zielsetzung des Wettbewerbs:** Es wird angeschaut, ob der Wettbewerb explizit auf unüberwachtes Lernen abzielt, insbesondere auf Clustering-Aufgaben. Wettbewerbe mit anderen Schwerpunkten, wie Klassifikation, LLM oder Regression, werden ausgeschlossen.
- **Datencharakteristik:** Es werden ausschließlich unbeschriftete und nicht zeitreihenbasierte Datensätze betrachtet, die unüberwachtes Lernen ermöglichen.
- **Bewertungsmetriken:** Es wird vorausgesetzt, dass Metriken zur Bewertung der Clusterqualität eingesetzt werden, beispielsweise der Silhouettenkoeffizient oder der Adjusted Rand Index.
- **Verfügbarkeit von Notebooks:** Ein Wettbewerb wird nur dann in die Analyse aufgenommen, wenn mindestens 50 öffentlich zugängliche Notebooks existieren.

C. Stichprobenziehung der Notebooks

Für jeden einzelnen Wettbewerb wird eine Stichprobe von 50 Notebooks gezogen. Diese Größe wurde gewählt, um eine repräsentative Analyse der angewandten Methoden zu gewährleisten, ohne den praktischen Rahmen der Untersuchung, insbesondere hinsichtlich Zeit und Rechenressourcen, zu sprengen. Die Festlegung der Stichprobengröße orientiert sich an etablierten Empfehlungen zur statistischen Power und Generalisierbarkeit von Forschungsergebnissen [13]–[15].

Die gezogene Stichprobe wird systematisch in einer Exceltabelle dokumentiert (siehe Abbildung 2). Diese Tabelle enthält folgende Informationen zu jedem Notebook:

- Name der Competition
- Anzahl der insgesamt verfügbaren Notebooks für die Competition
- Direktlink zum jeweiligen Notebook

Diese strukturierte Erfassung ermöglicht eine übersichtliche Verwaltung der Notebooks und bildet die Grundlage für die anschließende Analyse.

D. Analyse der Notebooks

Die ausgewählten Notebooks werden einer manuellen Analyse unterzogen, in der sowohl die Code-Segmente als auch die begleitenden Kommentare berücksichtigt werden. Im Mittelpunkt steht dabei die Identifikation der implementierten

Competition Slug	Total Notebooks	Notebook Link	Score	Algo
dmassign1	64	https://www.kaggle	0.40427	KMeans
dmassign1	64	https://www.kaggle	-	AgglomerativeClustering
dmassign1	64	https://www.kaggle	-	AgglomerativeClustering

Abbildung 2. Exceltabelle mit den gesammelten Stichprobennotebooks, inklusive Wettbewerb, Notebook-Link und Analyseergebnissen (eigene Darstellung).

Clustering-Algorithmen. Zusätzlich wird der in den Notebooks angegebene Score der jeweiligen Algorithmen erfasst, um deren relative Leistungsfähigkeit zu bewerten.

Die Ergebnisse dieser Analyse – also die identifizierten Algorithmen sowie deren zugehörige Scores – werden in der Exceltabelle (siehe Abbildung 2) dokumentiert. Dadurch wird eine systematische Auswertung der eingesetzten Methoden und ihrer Performance über verschiedene Wettbewerbe hinweg ermöglicht.

Anschließend werden die Häufigkeiten der verwendeten Clustering-Algorithmen ermittelt. Im weiteren Verlauf werden die fünf am häufigsten eingesetzten Algorithmen miteinander verglichen und anhand fiktiver Daten grafisch veranschaulicht, um ihre Unterschiede darzustellen.

V. Durchführung

Im Folgenden wird die praktische Umsetzung der in Abschnitt IV definierten Methodik beschrieben. Ziel war es, geeignete Kaggle-Wettbewerbe zu identifizieren, die als Plattform für die Analyse von Clustering-Methoden dienen. Anschließend wurden relevante Notebooks systematisch ausgewertet, um Erkenntnisse über die Verbreitung und Performance verschiedener Clustering-Algorithmen zu gewinnen.

A. Auswahl der Wettbewerbe

Zur Identifikation von Wettbewerben, die für die Analyse von Clustering-Methoden in Frage kommen, wurde die Such- und Filterfunktion von Kaggle direkt genutzt. Dabei kamen gezielt die Schlagwörter `Clustering` und `Unsupervised` zum Einsatz, um Wettbewerbe mit einem unüberwachten Lernfokus herauszufiltern. Zusätzlich wurde der Filter auf „relevanteste Ergebnisse“ eingestellt, sodass primär Wettbewerbe mit hoher Relevanz in den Resultaten auftauchten. Diese Suche lieferte **94 Treffer**, die anschließend anhand der im Methodikteil definierten zusätzlichen Auswahlkriterien weiter eingegrenzt wurden.

Letztlich konnten drei geeignete Clustering-Wettbewerbe für die weitere Analyse identifiziert werden:

- `tabular-playground-series-jul-2023`
- `bigdata-and-datamining-2nd-ex`
- `dmassign7`

Alle drei Wettbewerbe erfüllen die zuvor definierten Kriterien.

B. Stichprobenziehung und Datenaufbereitung

Zur Erfassung der 50 Notebooks pro Wettbewerb wurde ein Python-Skript entwickelt (siehe Kapitel V-E), um die Stichproben zu ziehen.

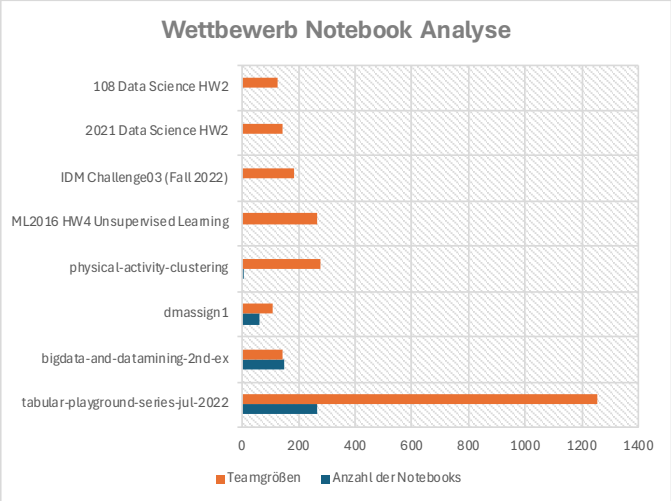

Abbildung 3. Anteil geeigneter Clustering-Wettbewerbe nach Filterkriterien (eigene Darstellung)

Anschließend wurden die Notebooks einzeln begutachtet und analysiert, welcher Algorithmus verwendet wurde. Dabei orientierte sich die Kategorisierung an den im Methodikteil definierten Kriterien.

Die Ergebnisse wurden systematisch in einer Excel-Tabelle dokumentiert, welche um folgende Spalten erweitert wurde:

- **Competition Slug:** Eindeutige Kennung des Wettbewerbs.
- **Total Notebooks:** Gesamtanzahl der im Wettbewerb vorhandenen Notebooks.
- **Notebook Link:** Direkter Link zum jeweiligen Notebook.
- **Score:** Erzielter Score des Modells (sofern vorhanden).
- **Algo:** Verwendeter Clustering-Algorithmus (z. B. anhand von Import-Anweisungen wie `sklearn.KMeans`).

Abbildung 4 zeigt einen Auszug aus der Excel-Tabelle, in der neben den einzelnen Notebook-Einträgen auch aggregierte Ergebnisse je Wettbewerb zusammengefasst wurden.

Competition Slug	Total Notebooks	Notebook Link	Score	Algo
dmassign1	64	https://www.kag	0.40427	KMeans
dmassign1	64	https://www.kag	-	AgglomerativeClustering
dmassign1	64	https://www.kag	-	AgglomerativeClustering
dmassign1	64	https://www.kag	-	-
dmassign1	64	https://www.kag	0.40478	KMeans
dmassign1	64	https://www.kag	-	-
dmassign1	64	https://www.kag	-	KMeans
dmassign1	64	https://www.kag	-	AgglomerativeClustering
dmassign1	64	https://www.kag	-	KMeans
dmassign1	64	https://www.kag	-	AgglomerativeClustering
dmassign1	64	https://www.kag	-	KMeans

Abbildung 4. Auszug der Excel-Tabelle mit den aggregierten Notebook-Daten

C. Analyse der Clustering-Algorithmen

Über alle Wettbewerbe hinweg ergab die Analyse der Excel-Daten, dass der Algorithmus **KMeans** am häufigsten verwendet wurde.

Wie in Abbildung 5 dargestellt, war **KMeans** der am häufigsten verwendete Algorithmus, gefolgt von **Agglomerative Clustering** und **Spectral Clustering**.

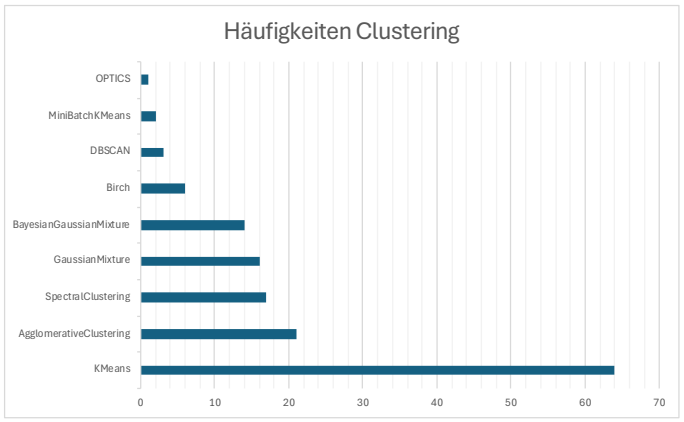

Abbildung 5. Häufigkeit der verwendeten Clustering-Algorithmen in den analysierten Notebooks

Weitere verbreitete Verfahren sind **Gaussian Mixture Models (GMM)**, **Bayesian Gaussian Mixture** sowie **Birch**.

Die Algorithmen **DBSCAN** und **OPTICS**, die insbesondere für dichtebasierte Clustering-Methoden bekannt sind, wurden seltener verwendet. Dies könnte darauf hindeuten, dass die in den Kaggle-Wettbewerben verwendeten Datensätze sich besser für klassische partitionierende oder hierarchische Verfahren eignen.

D. Scoring und Performance der Clustering-Algorithmen

Die Performance der Algorithmen variiert deutlich je nach Wettbewerb:

dmassign-Wettbewerbe: **KMeans** wurde am häufigsten eingesetzt und erreichte einen Score von **0.4** (23 Notebooks). Andere Algorithmen wie Agglomerative Clustering, Birch, MiniBatchKMeans sowie dichtebasierte Methoden (DBSCAN, OPTICS) schnitten schlechter ab.

bigdata-and-datamining-2nd-ex: Hier dominierte **Birch** (Score 0.96), gefolgt von Agglomerative Clustering (0.95), Spectral Clustering (0.89) und KMeans (0.87). Auch GMM (0.47) und MiniBatchKMeans (0.84) zeigten gute Ergebnisse, während DBSCAN und OPTICS wieder schwach abschnitten.

tabular-playground-series-jul: Probabilistische Modelle waren vorteilhaft: **Bayesian Gaussian Mixture** erzielte einen Score von **0.49**, gefolgt von Gaussian Mixture (0.4) und KMeans (0.3). Andere Ansätze wie Agglomerative Clustering, Spectral Clustering und Birch erzielten kaum signifikante Ergebnisse.

1) Kurzes Fazit:

- **KMeans** wurde am häufigsten eingesetzt, erzielte aber nur in bestimmten Wettbewerben Spitzenwerte.
- **Birch** und Agglomerative Clustering überzeugten im *bigdata-and-datamining-2nd-ex*-Wettbewerb.
- Probabilistische Modelle (GMM, Bayesian Gaussian Mixture) waren im *tabular-playground-series-jul* Wettbewerb vorteilhaft.
- Dichtebasierte Methoden (DBSCAN, OPTICS) schnitten durchgehend schlecht ab.

E. Code Implementierung

In dieser Arbeit wurde ein Python-Skript entwickelt, das die Kaggle-API nutzt, um automatisch Notebooks aus verschiedenen Wettbewerben abzurufen und in einer Excel-Datei zusammenzufassen. Zunächst werden die benötigten Bibliotheken importiert und die Authentifizierung für die Kaggle-API durchgeführt. Ein Dictionary speichert die URLs der Wettbewerbe und die zugehörigen Slugs, um die API-Abfragen zu erleichtern. Das Skript erstellt einen Ausgabeordner für die Excel-Datei. Für jeden Wettbewerb im Dictionary werden mittels Pagination (Aufteilung der Daten in mehrere Seiten) alle verfügbaren Notebooks abgerufen, wobei pro Seite bis zu 100 Notebooks angefordert werden. Fehler wie ein 404-Fehler bei fehlendem Zugriff oder nicht vorhandenen Wettbewerben werden abgefangen und entsprechend behandelt. Nach der Erfassung der Notebooks für einen Wettbewerb wählt das Skript zufällig bis zu 50 Notebooks aus. Für jedes dieser Notebooks werden Informationen wie die Wettbewerbs-URL, der Wettbewerb-Slug, die Gesamtzahl der gefundenen Notebooks sowie der direkte Link zum jeweiligen Notebook extrahiert und in einer Liste gespeichert. Abschließend werden die gesammelten Daten in ein Pandas DataFrame überführt und als Excel-Datei im definierten Ausgabeordner gespeichert.

Weitere Details und den vollständigen Code ist im GitHub-Repository[1] zu finden.

VI. Clustering-Algorithmen

In diesem Kapitel werden die fünf am häufigsten verwendeten Clustering-Methoden vorgestellt und miteinander verglichen.

Das in den folgenden Abschnitten verwendete Datenset ist im verlinkten Repository verfügbar. [2]

A. K-Means

Der **K-Means-Algorithmus** ist ein weit verbreitetes, partitionierendes Verfahren zur Gruppierung von Datenpunkten in Cluster. Ziel ist es, eine gegebene Menge von Datenpunkten in eine im Voraus festgelegte Anzahl k von Clustern zu unterteilen, sodass die Punkte innerhalb eines Clusters möglichst homogen sind, während die Unterschiede zwischen den Clustern maximiert werden. [16]

1) Funktionsweise des Algorithmus: Der Algorithmus startet mit der Bestimmung der gewünschten Clusteranzahl k. Anschließend werden zufällig k Startpunkte – die sogenannten Zentroiden – gewählt, welche vorläufig als Mittelpunkte der Cluster fungieren [16]. Danach erfolgt in zwei Hauptschritten ein iterativer Prozess:

1) **Zuordnung:** Jeder Datenpunkt wird dem nächstgelegenen Zentroiden (gemessen an der euklidischen Distanz) zugewiesen, wodurch erste Cluster entstehen.
2) **Neuberechnung:** Für jedes Cluster wird ein neuer Zentroid berechnet, indem der Mittelwert aller ihm zugeordneten Datenpunkte bestimmt wird:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Diese beiden Schritte werden solange wiederholt, bis sich die Positionen der Zentroiden kaum noch verändern – ein Zustand, der als Konvergenz bezeichnet wird. [16]

2) Anwendung auf verschiedene Datensätze: Im Folgenden wird das Verhalten des K-Means-Algorithmus an drei unterschiedlichen Datensätzen analysiert. Die Ergebnisse werden jeweils durch zugehörige Abbildungen illustriert.

a) 1. Datensatz mit klar abgegrenzten Clustern: Der erste Datensatz zeigt eine reguläre Verteilung von Punkten, die in klar getrennte Cluster gruppiert sind. Aufgrund der eindeutigen Clusterstruktur gelingt es K-Means hier, die Clusterzentren präzise zu bestimmen und die Datenpunkte korrekt zuzuordnen. Abbildung 6 verdeutlicht diese Situation, wobei der Silhouette-Koeffizient mit einem Wert von 0.41 auf eine insgesamt moderate Clusterqualität hinweist.

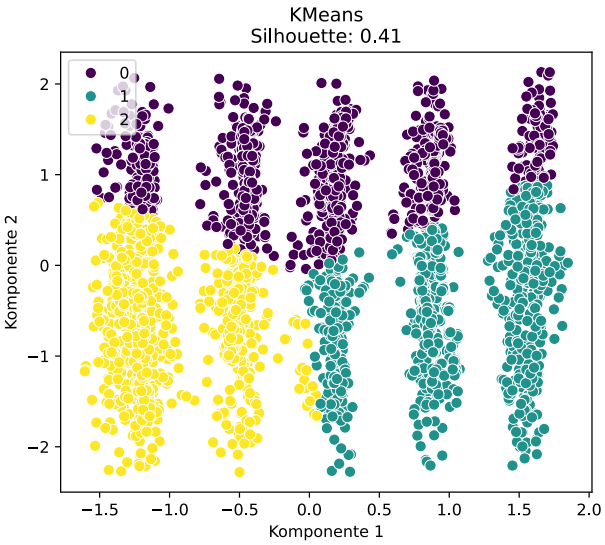

Abbildung 6. Ergebnisse des K-Means-Clustering auf einem Datensatz mit klar abgegrenzten Clustern.

b) 2. Datensatz mit Ausreißern: Im zweiten Szenario kommen zu den regulären Clustern einige Ausreißer hinzu. Obwohl diese zusätzlichen Punkte die Berechnung der Zentroiden leicht verzerren können, ist es dem Algorithmus dennoch möglich, eine sinnvolle Clusterstruktur zu erzeugen. Wie in Abbildung 7 zu sehen, führt dieser Datensatz zu einem etwas höheren Silhouette-Koeffizienten von 0.53. Dies legt nahe, dass der Großteil der Datenpunkte trotz der Ausreißer in gut definierte Cluster eingeteilt wird, während die Ausreißer nur einen begrenzten Einfluss haben.

c) 3. Datensatz mit Spiralstruktur: Der dritte Datensatz stellt eine besondere Herausforderung dar, da er eine nichtkonvexe, spiralartige Anordnung der Datenpunkte aufweist. Da K-Means auf euklidischer Distanz basiert und grundsätzlich

[1] https://github.com/XplorodoX/BigData
[2] https://www.kaggle.com/datasets/joonasyoon/clustering-exercises

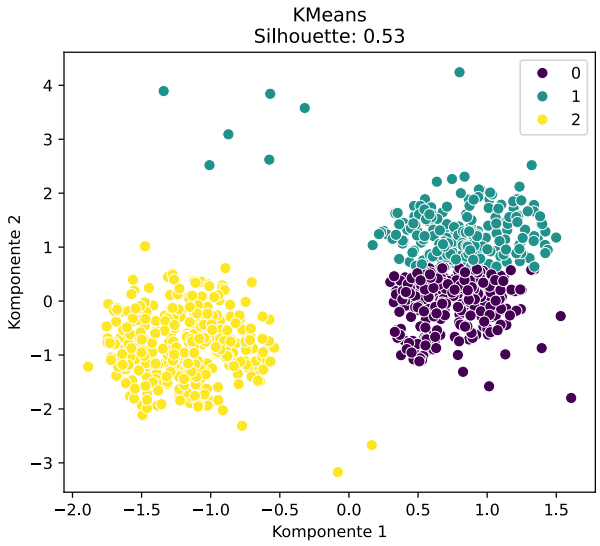

Abbildung 7. K-Means-Clustering eines Datensatzes mit Ausreißern.

isotrope (rundliche) Cluster bevorzugt, kann der Algorithmus die komplexe, nicht-lineare Struktur der Spirale nicht adäquat erfassen. Abbildung 8 zeigt, dass die resultierenden Clustergrenzen suboptimal sind. Obwohl der Silhouette-Koeffizient auch hier mit 0.41 bewertet wird, spiegelt dieser Wert nicht die tatsächliche Unzulänglichkeit der Clusterzuordnung in Bezug auf die zugrunde liegende Spiralstruktur wider.

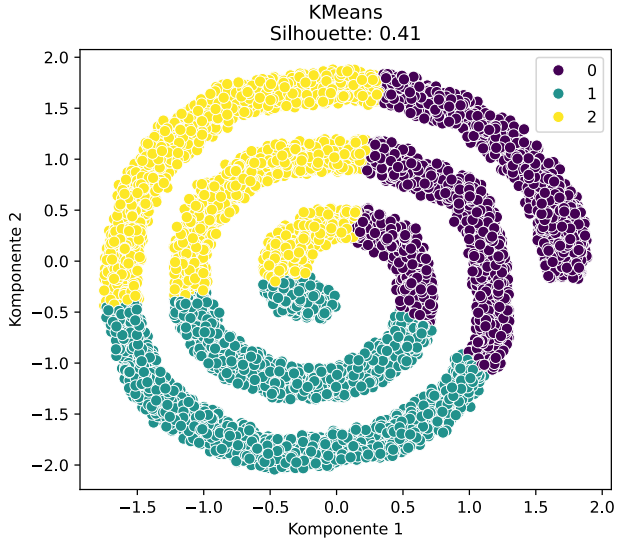

Abbildung 8. Ergebnisse des K-Means-Clustering auf einem Datensatz mit spiralartiger Verteilung.

B. Spectral Clustering

Eine der populärsten Methoden für Graph Clustering ist das **Spectral Clustering**, das auf der Spektralanalyse der Adjazenzmatrix oder Laplace-Matrix eines Graphen basiert. [17]

Im Gegensatz zu anderen Clustering-Verfahren wie *K-Means* nutzt Spectral Clustering die Eigenwerte und Eigenvektoren einer Matrix zur Transformation der Daten in einen neuen Raum, in dem eine einfachere Trennung der Cluster möglich ist. [17]

1) Funktionsweise des Algorithmus: Spectral Clustering kann in drei Hauptschritten zusammengefasst werden:

1) **Graph-Konstruktion:** Aus den gegebenen Daten wird ein Graph $G = (V, E)$ erstellt, in dem Knoten die Datenpunkte und Kanten die Ähnlichkeiten zwischen den Punkten repräsentieren [17]. Die Ähnlichkeit kann beispielsweise durch eine Distanzfunktion wie die Gauss-Kernel-Funktion definiert werden:
$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right).$$
2) **Spektrale Transformation:** Basierend auf dem Graphen wird eine Matrix (z. B. die normalisierte Laplace-Matrix L_{sym}) berechnet:
$$L_{sym} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}.$$
Anschließend werden die ersten k Eigenvektoren von L_{sym} als neue Repräsentation der Datenpunkte genutzt. [17]
3) **K-Means Clustering:** Die transformierten Datenpunkte werden mit dem *K-Means*-Algorithmus in k Cluster unterteilt. [17]

2) Analyse der Clustering-Ergebnisse: In diesem Abschnitt werden die Ergebnisse des Spectral Clustering, angewendet auf drei unterschiedliche Datensätze, detailliert untersucht.

a) Grundlegende Clusterstruktur: Abbildung 18 zeigt ein einfaches Clustermuster, bei dem drei Cluster klar voneinander getrennt sind. Der erreichte Silhouette-Score von 0,22 deutet auf eine moderate Clustertrennung hin. Die Cluster wirken relativ kompakt und passen gut zur zugrunde liegenden Datenstruktur. Diese Darstellung bestätigt, dass der Algorithmus auch in Szenarien mit überschaubaren und gut abgegrenzten Gruppen sinnvolle Ergebnisse liefert.

b) Cluster mit Ausreißern: In Abbildung 19 wird ein Datensatz präsentiert, der zusätzlich einige Ausreißer beinhaltet, welche sich in deutlich entfernten Bereichen des Merkmalsraums befinden. Trotz dieser Störeinflüsse steigt der Silhouette-Score hier auf 0,52, was darauf schließen lässt, dass die Hauptcluster robust und klar voneinander getrennt bleiben. Die Grafik illustriert, dass der Algorithmus in der Lage ist, die zentrale Clusterstruktur zu erkennen, auch wenn vereinzelte Datenpunkte außerhalb des typischen Verteilungsbereichs liegen.

c) Spiralförmige Clusterstruktur: Abbildung 20 veranschaulicht eine komplexe, nichtlineare Clusterstruktur mit einer spiralartigen Verteilung der Datenpunkte. Mit einem Silhouette-Score von lediglich 0,10 wird deutlich, dass die Cluster hier stark überlappen und weniger eindeutig getrennt sind. Diese Darstellung unterstreicht die Grenzen des Spectral Clustering, insbesondere bei der Erkennung von nicht-euklidischen und kompliziert strukturierten Datenverteilungen.

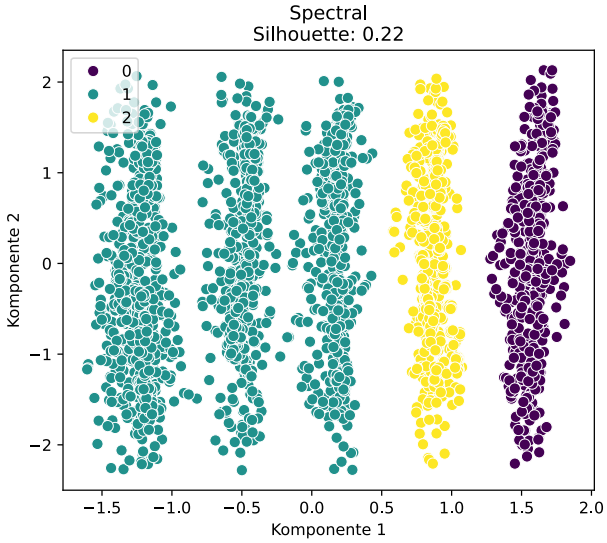

Abbildung 9. Grundlegende Clusterstruktur: Drei klar abgegrenzte Cluster mit einem Silhouette-Score von 0,22.

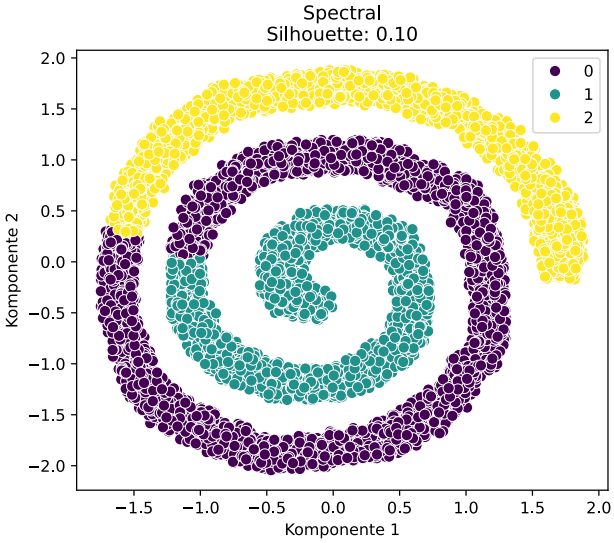

Abbildung 11. Spiralförmige Clusterstruktur: Die komplexe, spiralartige Anordnung der Datenpunkte führt zu einem niedrigen Silhouette-Score von 0,10.

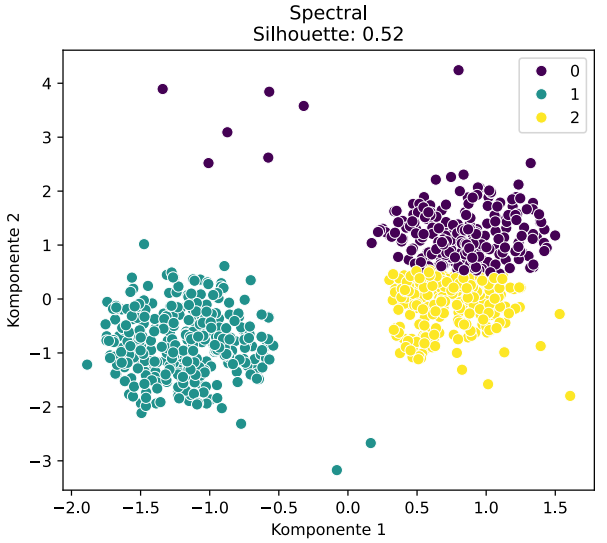

Abbildung 10. Cluster mit Ausreißern: Trotz vorhandener Ausreißer wird eine robuste Trennung der Hauptcluster erreicht (Silhouette-Score: 0,52).

C. Agglomerative Clustering

Hierarchisches agglomeratives Clustering (HAC) ist eine fundamentale Technik im Bereich des *Unsupervised Learning*. Es gehört zu den sequentiellen, hierarchischen, nicht überlappenden Methoden (SAHN) und beginnt mit einzelnen Punkten als Cluster, die iterativ zu größeren Clustern zusammengeführt werden. [18]

1) Grundprinzip des Algorithmus: Der Algorithmus beginnt mit einer Menge von N Datenpunkten, wobei jeder Punkt als ein einzelner Cluster betrachtet wird. Danach erfolgt eine schrittweise Zusammenführung der Cluster basierend auf einer Distanzenmatrix [18]. Die Hauptschritte sind:

1) **Distanzberechnung**: Eine Distanzenmatrix D mit paarweisen Distanzen wird berechnet. [18]
2) **Cluster-Zusammenführung**: Die beiden Cluster mit der geringsten Distanz werden zusammengeführt. [18]
3) **Distanzen-Update**: Die Distanzen der neuen Cluster zu den verbleibenden Clustern werden berechnet. [18]
4) **Wiederholung**: Schritte 2 und 3 werden wiederholt, bis nur noch ein Cluster übrig bleibt. [18]

2) Analyse des Agglomerativen Clusterings: Im Folgenden wird die Anwendung des Agglomerativen Clusterings auf drei unterschiedliche Datensätze detailliert analysiert.

a) Basis-Datensatz: Beim Basis-Datensatz ergibt das Agglomerative Clustering eine relativ klare Trennung der Cluster. Die Clusterzentren sind gleichmäßig verteilt, was durch einen Silhouette-Wert von 0.36 bestätigt wird – ein Indikator für eine moderate Clusterqualität. Dennoch sind an den Rändern benachbarter Cluster teilweise Überlappungen zu erkennen, was darauf hindeutet, dass die Trennschärfe nicht vollkommen gegeben ist. In Abbildung 18 ist das Clustering-Ergebnis des Basis-Datensatzes dargestellt.

b) Datensatz mit Ausreißern: Wird der Algorithmus auf einen Datensatz mit Ausreißern angewandt, zeigt sich, dass das Agglomerative Clustering empfindlich auf stark abweichende Daten reagiert. Einzelne, weit entfernte Punkte wirken sich signifikant auf die Clusterbildung aus, sodass einige Cluster gestreckt und inhomogen erscheinen. Ein erhöhter Silhouette-Wert von 0.52 signalisiert zwar eine insgesamt stärkere Trennung der Cluster, jedoch deuten die verzerrten Formen auf eine ungleichmäßige Datenverteilung hin. Dieses Verhalten wird in Abbildung 19 deutlich, wo das Clustering-Ergebnis des Datensatzes mit Ausreißern veranschaulicht wird.

c) Spiralförmiger Datensatz: Der spiralförmige Datensatz stellt den Algorithmus vor besondere Herausforderungen. Aufgrund der verschlungenen Struktur der Daten werden benachbarte Punkte häufig fälschlicherweise demselben Cluster zugeordnet, obwohl sie global betrachtet nicht zusammengehören. Dies spiegelt sich in einem Silhouette-Wert

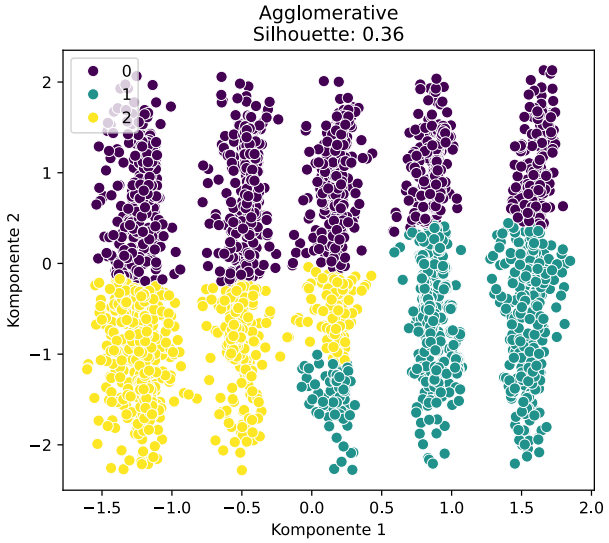

Abbildung 12. Clustering-Ergebnis für den Basis-Datensatz.

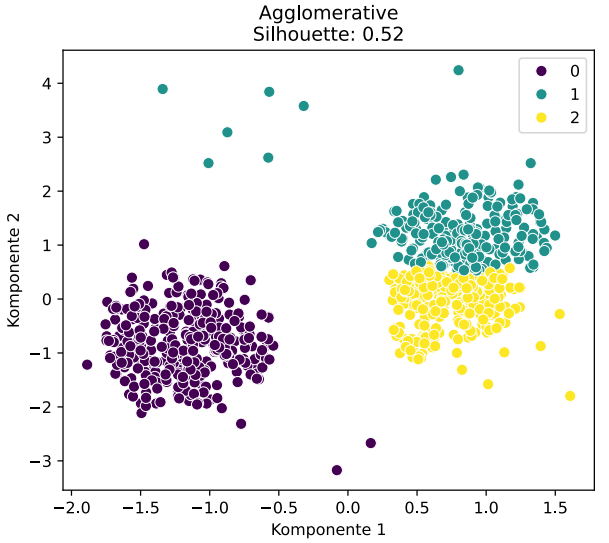

Abbildung 13. Clustering-Ergebnis für den Datensatz mit Ausreißern.

von 0.38 wider und unterstreicht die Schwierigkeiten des Agglomerativen Clusterings bei der Erkennung nicht-konvexer Strukturen. Abbildung 20 illustriert eindrucksvoll, wie die komplexe Geometrie der Daten zu einer weniger eindeutigen Clusterbildung führt.

D. Gaussian Mixture

Das **Gaussian Mixture Model** (GMM) ist ein probabilistisches Modell, das annimmt, dass die Datenpunkte von einer Mischung einer endlichen Anzahl von Gaußschen Verteilungen mit unbekannten Parametern generiert wurden. GMM kann als Verallgemeinerung von k-Means betrachtet werden, da es nicht nur die Clusterzentren, sondern auch die Kovarianzstruktur der Daten berücksichtigt. [19]

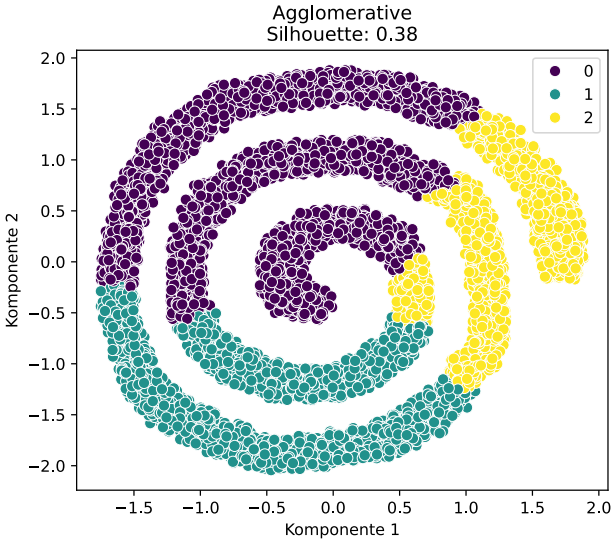

Abbildung 14. Clustering-Ergebnis für den spiralförmigen Datensatz.

1) Mathematische Definition: Ein GMM mit K Komponenten ist definiert als:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{1}$$

wobei:

- π_k die Mischungsgewichte mit $\sum_{k=1}^{K} \pi_k = 1$ sind.
- $\mathcal{N}(x|\mu_k, \Sigma_k)$ die multivariate Gaußsche Verteilung mit Mittelwert μ_k und Kovarianzmatrix Σ_k darstellt. [19]

Die Parameterschätzung erfolgt mittels der **Expectation-Maximization** (EM)-Methode. [19]

2) Expectation-Maximization Algorithmus: Die EM-Iteration besteht aus zwei Schritten:

- **E-Schritt**: Berechnung der Verantwortlichkeiten γ_{ik} für jede Komponente:

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)} \tag{2}$$

- **M-Schritt**: Aktualisierung der Parameter:

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{3}$$

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} x_i \tag{4}$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T \tag{5}$$

wobei $N_k = \sum_{i=1}^{N} \gamma_{ik}$ die gewichtete Anzahl der Punkte im k-ten Cluster ist. [19]

3) Analyse der Clustering-Ergebnisse mittels Gaussian Mixture Model: Im Folgenden wird der Einsatz des Gaussian Mixture Models (GMM) zur Clusteranalyse an Datensätzen mit unterschiedlichen Strukturen anhand von drei Abbildungen veranschaulicht. Zur Bewertung der Clustering-Qualität wurde der Silhouettenkoeffizient herangezogen, dessen Werte je

nach Datenverteilung variieren und somit Aufschluss über die Anpassungsfähigkeit des Modells in den jeweiligen Szenarien geben.

a) 1. Einfache Clusterstruktur: Abbildung 15 zeigt die Anwendung des GMM auf einen Datensatz mit einer klar abgegrenzten, einfachen Clusterstruktur. Der errechnete Silhouettenwert von 0.41 weist auf eine moderate Clustertrennung hin. Obwohl die Cluster grundsätzlich voneinander unterscheidbar sind, lassen sich in der Darstellung leichte Überlappungen erkennen, was auf natürliche Streuungen innerhalb der Daten hindeutet. Die Abbildung illustriert, wie das Modell die Clusterzentren und zugehörigen Kovarianzmatrizen optimal an die Verteilung der Daten anpasst.

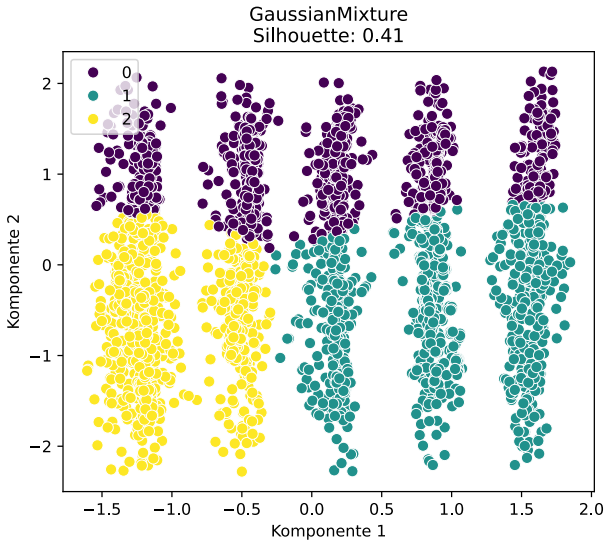

Abbildung 15. Clusteranalyse mittels GMM an einem Datensatz mit einfacher, klar abgegrenzter Clusterstruktur. Die Darstellung zeigt die Anpassung der Clusterzentren und Kovarianzmatrizen an die Datenverteilung, was zu einem Silhouettenwert von 0.41 führt.

b) 2. Datensatz mit Ausreißern: In Abbildung 16 wird das GMM auf einen Datensatz angewandt, der neben den Hauptclustern auch einige Ausreißer enthält. Hier zeigt sich ein deutlich höherer Silhouettenwert von 0.61, was auf eine klarere Trennung der Kerncluster hinweist. Dies liegt darin begründet, dass das Modell den Einfluss der Ausreißer weitgehend minimiert und robuste Clusterzentren ermittelt. Dennoch sind in der Abbildung einzelne Punkte außerhalb der dominierenden Cluster erkennbar, was potenziell zu leichten Verzerrungen der Modellparameter führen kann.

c) 3. Komplexe, nicht-lineare Clusterstruktur: Abbildung 17 veranschaulicht den Einsatz des GMM auf einen Datensatz mit einer spiralartigen, nicht-linearen Clusterstruktur. Der hier beobachtete Silhouettenwert von 0.40 signalisiert, dass das Modell Schwierigkeiten hat, die komplexe Form der Cluster adäquat abzubilden. Da das GMM von der Annahme ausgeht, dass die Daten aus einer Mischung von gaußförmigen (ellipsenförmigen) Verteilungen stammen, kommt es in diesem Fall zu einer unzureichenden Modellierung, was sich in überlappenden Clustergrenzen widerspiegelt. Die Abbildung verdeutlicht, dass die nicht-linearen Strukturen der Daten

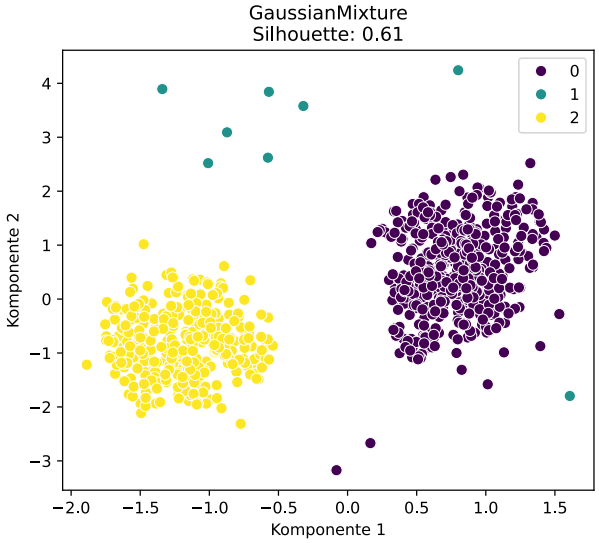

Abbildung 16. Ergebnis der GMM-Anwendung auf einen Datensatz mit Ausreißern. Trotz der Extremwerte wird eine klare Clustertrennung erreicht, was sich in einem erhöhten Silhouettenwert von 0.61 widerspiegelt.

durch die klassischen Annahmen des GMM nur schwer zu erfassen sind.

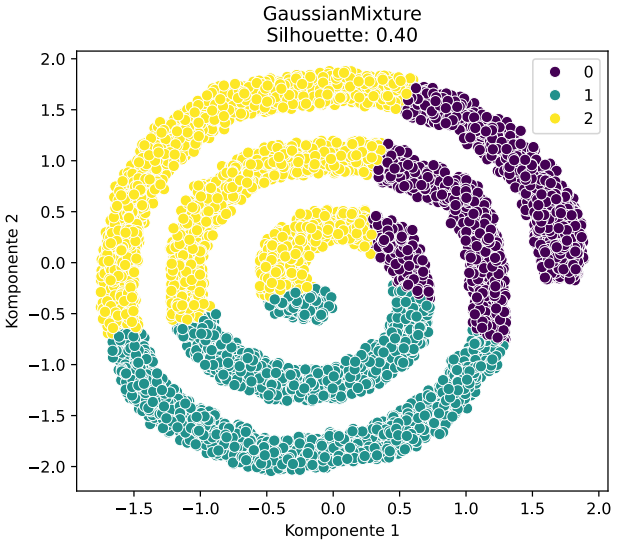

Abbildung 17. Analyse eines Datensatzes mit spiralartiger, nicht-linearer Clusterstruktur mittels GMM. Die unzureichende Anpassung des Modells an die komplexe Form führt zu einem niedrigen Silhouettenwert von 0.40 und überlappenden Clustergrenzen.

E. Bayesian Gaussian Mixture

Das **Bayesian Gaussian Mixture Model** (BGMM) ist eine probabilistische Erweiterung des klassischen Gaussian Mixture Models (GMM), das bayesianische Methoden verwendet, um eine flexiblere Anzahl von Komponenten zu erlauben. BGMM basiert auf der **variationalen Bayes-Inferenz** und verwendet Dirichlet-Prozesse als Prior für die Mischungsgewichte. [20]

1) Mathematische Definition: Ein Bayesian Gaussian Mixture Model mit K Komponenten ist definiert als:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{6}$$

wobei:

- π_k die Mischungsgewichte sind, die aus einer Dirichlet-Verteilung gezogen werden. [20]
- $\mathcal{N}(x|\mu_k, \Sigma_k)$ eine multivariate Gaußsche Verteilung mit Mittelwert μ_k und Kovarianzmatrix Σ_k ist. [20]

Zusätzlich werden Priorverteilungen auf die Parameter π_k, μ_k und Σ_k definiert:

- $\pi_k \sim \text{Dirichlet}(\alpha)$ (Dirichlet-Prozess für Gewichte). [20]
- $\mu_k \sim \mathcal{N}(\mu_0, \lambda_0^{-1}\Sigma_k)$. [20]
- $\Sigma_k^{-1} \sim \text{Wishart}(W_0, \nu_0)$ (Inverse-Wishart als Prior für Kovarianz). [20]

2) Variationale Bayes-Inferenz: Statt der klassischen **Expectation-Maximization** (EM)-Methode wie bei GMM wird für BGMM die **variationale Bayes-Inferenz** (VB) verwendet. Diese führt eine Regularisierung durch Integration von Priorwissen durch. [20]

Die Updates für die Parameter erfolgen durch:

$$\tilde{\pi_k} = \frac{\alpha_k + N_k}{\sum_{j=1}^{K}(\alpha_j + N_j)} \tag{7}$$

$$\tilde{\mu_k} = \frac{\lambda_0 \mu_0 + N_k \bar{x}_k}{\lambda_0 + N_k} \tag{8}$$

$$\tilde{\Sigma_k}^{-1} = W_0 + \sum_{i=1}^{N_k}(x_i - \tilde{\mu_k})(x_i - \tilde{\mu_k})^T \tag{9}$$

Hierbei ist N_k die effektive Anzahl von Punkten im Cluster k und $\bar{x}_k$

3) Analyse der Clustering-Ergebnisse: Zur Untersuchung des Verhaltens des Bayesian Gaussian Mixture Model (BGMM) wurden drei verschiedene Datensätze verwendet: ein einfacher, wohlstrukturierter Datensatz, ein Datensatz mit Ausreißern und ein spiralartig angeordneter Datensatz. Die Clustering-Ergebnisse wurden anhand der Silhouette-Werte bewertet, die als Maß für die Cluster-Qualität dienen.

a) Grundlegendes Clustering: Im ersten Szenario, dargestellt in Abbildung 18, zeigt das BGMM eine moderate Clustering-Leistung mit einem Silhouette-Wert von 0.35. Die Cluster erscheinen zwar getrennt, jedoch gibt es leichte Unschärfen an den Grenzen. Dies deutet darauf hin, dass die Cluster teilweise überlappen oder dass einige Datenpunkte nicht eindeutig einem Cluster zugeordnet werden können.

b) Clustering mit Ausreißern: Das zweite Szenario enthält Ausreißer, was in Abbildung 19 zu sehen ist. Der Silhouette-Wert beträgt 0.62, was darauf hinweist, dass die Cluster in diesem Fall besser getrennt sind als im vorherigen Szenario. Dies könnte darauf hindeuten, dass das Modell die Ausreißer entweder effektiv als separate Cluster erkennt oder dass die Hauptcluster weniger überlappen. Die erhöhte Cluster-Qualität deutet darauf hin, dass BGMM durch seine probabilistische Modellierung robuster gegenüber solchen Störungen sein kann.

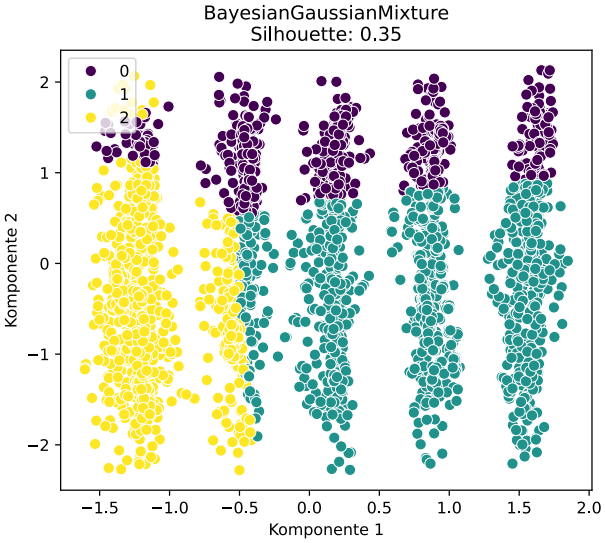

Abbildung 18. Clustering-Ergebnis für einen einfachen Datensatz mit BGMM.

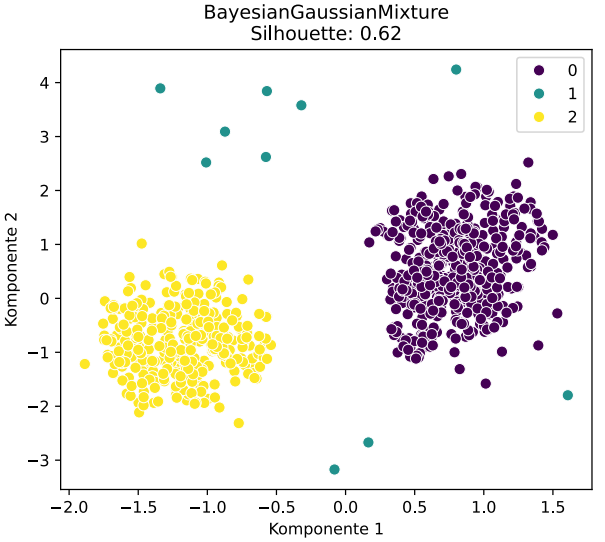

Abbildung 19. Clustering-Ergebnis mit BGMM für einen Datensatz mit Ausreißern.

c) Clustering einer Spiralstruktur: Im dritten Szenario wurde eine komplexere, nicht-konvexe Struktur untersucht, wie in Abbildung 20 dargestellt. Der Silhouette-Wert ist mit 0.30 am niedrigsten, was darauf hindeutet, dass das Modell Schwierigkeiten hat, die nicht-linearen Clusterformen korrekt zu erfassen. Dies ist erwartungsgemäß, da BGMM auf einer Gaußschen Modellierung basiert, die am besten für elliptische Cluster geeignet ist. Die Spiralstruktur stellt daher eine Herausforderung dar, und es ist wahrscheinlich, dass sich Cluster überlappen oder nicht korrekt erkannt werden.

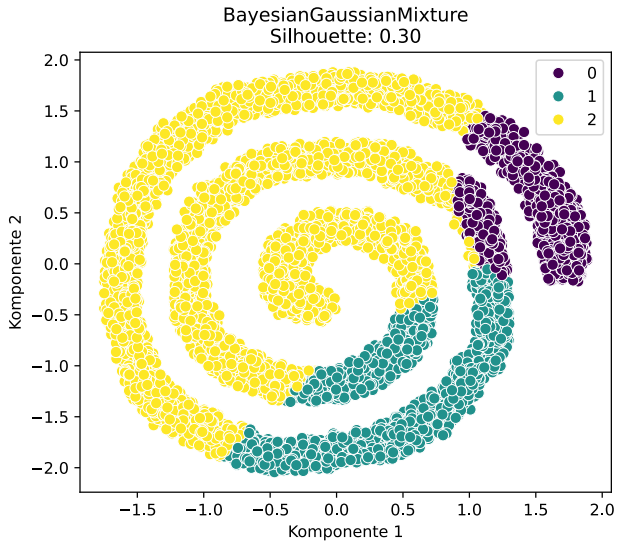

Abbildung 20. Clustering-Ergebnis mit BGMM für eine Spiralstruktur.

VII. Vergleich der Clustering-Algorithmen – Eine differenzierte Betrachtung

Hier wird auf die Clustering Algos nochmal im Vergleich zueinander eingegangen.

A. Übergreifende Eigenschaften und Herausforderungen

Das **K-Means**-Verfahren zeichnet sich durch eine hervorragende Rechengeschwindigkeit und eine einfache Implementierung aus, weshalb es häufig für große, gut separierbare Datensätze verwendet wird. Allerdings ist K-Means stark abhängig von der Initialisierung und der vorgegebenen Clusteranzahl. Zudem führt die strikte Fokussierung auf euklidische Distanzen zu Problemen, wenn es um nicht-konvexe oder ungleich verteilte Daten geht, da einzelne Ausreißer die Berechnung der Zentroiden erheblich verzerren können. [16]

Im Gegensatz dazu ermöglicht das **Spectral Clustering** durch die Transformation der Daten in einen spektralen Raum die Abbildung komplexer, nicht-linearer Strukturen, die mit klassischen Distanzmaßen nicht adäquat erfasst werden können. Auf diese Weise lassen sich Clusterformen erkennen, die über konvexe Strukturen hinausgehen. Allerdings ist dieser Ansatz mit einem hohen Rechenaufwand verbunden, da es notwendig ist, eine Ähnlichkeitsmatrix zu erstellen und eine Eigenwertzerlegung durchzuführen. Zudem hängt die Qualität der Ergebnisse stark von der Wahl des Ähnlichkeitsmaßes und der Skalierung der Daten ab. [17]

Das **Agglomerative Clustering** ist ein hierarchisches Verfahren, das den Vorteil bietet, dass keine vorherige Festlegung der Clusteranzahl erforderlich ist. Außerdem liefert es eine Baumstruktur (Dendrogramm), die zusätzliche Einsichten in die hierarchische Organisation der Daten ermöglicht. Allerdings reagiert dieser Ansatz empfindlich auf Ausreißer und lokale Verzerrungen. Die Ergebnisse können zudem erheblich mit der gewählten Linkage-Methode (z. B. Single, Complete oder Average) variieren, und bei sehr großen Datensätzen steigt der Rechenaufwand erheblich. [18]

Das **Gaussian Mixture Model (GMM)** basiert auf einer probabilistischen Modellierung, die es erlaubt, überlappende und ellipsoidale Cluster zu beschreiben. Neben den Clusterzentren werden hierbei auch die Varianzen innerhalb der Daten berücksichtigt, was zu einer flexibleren Modellierung führt. Allerdings schränkt die Annahme, dass alle Cluster einer Gauß-Verteilung folgen, die Anpassungsfähigkeit an stark nicht-lineare Strukturen ein. [19]

Das **Bayesian Gaussian Mixture Model (BGMM)** erweitert das klassische GMM durch die Einbeziehung bayesianischer Inferenz, was die Modellflexibilität insbesondere in Situationen erhöht, in denen die exakte Anzahl der Cluster unklar ist. Durch die probabilistische Regularisierung erweist sich BGMM als robuster gegenüber Ausreißern und verhindert eine Überanpassung, da Unsicherheiten explizit modelliert werden. Allerdings führt die zusätzliche Komplexität der variationalen Bayes-Inferenz zu einem höheren Rechenaufwand, und die Modellleistung kann stark von der Wahl der Priorparameter abhängen, sodass eine sorgfältige Abstimmung erforderlich ist. [20]

B. Leistungsfähigkeit in unterschiedlichen Datenszenarien

Die Eignung der verschiedenen Clustering-Algorithmen variiert in Abhängigkeit von der Struktur der vorliegenden Daten. Bei Datensätzen mit klaren, konvexen Clustern liefern klassische Verfahren wie K-Means, GMM und BGMM in der Regel sehr gute Ergebnisse. Insbesondere punktet K-Means durch seine Schnelligkeit, während die probabilistischen Ansätze darüber hinaus statistische Unsicherheiten modellieren können.

In Datensätzen, die durch Ausreißer gekennzeichnet sind, können deterministische Methoden wie K-Means und agglomerative Ansätze durch extreme Werte verzerrt werden. In solchen Fällen erweisen sich spectral-basierte sowie probabilistische Methoden als robuster, wobei BGMM durch seine Regularisierung besonders gut mit Ausreißern umgehen kann. Für komplexe, nicht-lineare Strukturen bieten spectral-basierte Methoden und hierarchische Ansätze klare Vorteile, da sie nicht auf die Annahme konvexer Formen beschränkt sind. Obwohl GMM und BGMM durch ihre probabilistische Modellierung eine gewisse Flexibilität aufweisen, stoßen sie bei stark nicht-linearen Verteilungen an ihre Grenzen, da die zugrunde liegende Annahme einer gaußförmigen Verteilung nicht immer gerechtfertigt ist.

C. Gesamteinschätzung und anwendungsspezifische Empfehlungen

Zusammenfassend hängt die Wahl des optimalen Clustering-Algorithmus entscheidend von den spezifischen Anforderungen des jeweiligen Anwendungsfalls ab. Für sehr große und überwiegend konvexe Datensätze bietet K-Means aufgrund seiner Einfachheit und Effizienz oft die beste Performance. Bei komplexen oder nicht-linearen Strukturen, bei denen herkömmliche Distanzmaße versagen, kann Spectral Clustering bessere Ergebnisse liefern, wenngleich dies mit einem höheren Rechenaufwand verbunden ist. Wenn explorative Analysen gewünscht werden, bei denen auch die hierarchische Struktur

der Daten von Interesse ist, eignet sich Agglomerative Clustering besonders gut, auch wenn hierbei die Sensitivität gegenüber Ausreißern beachtet werden muss. Für Anwendungen, die eine probabilistische Interpretation und eine Abschätzung von Unsicherheiten erfordern, sind GMM und insbesondere BGMM zu empfehlen, sofern der erhöhte Rechenaufwand und die Notwendigkeit einer sorgfältigen Parametrisierung in Kauf genommen werden können.

VIII. Fazit

Diese Arbeit veranschaulicht, dass kein einzelner Clustering-Algorithmus eine *Allzwecklösung* darstellt. Zwar zeigt sich **K-Means** aufgrund seiner einfachen Implementierung und geringen Rechenkosten als „Arbeitspferd“, das in vielen Datenszenarien solide Ergebnisse liefert. Dennoch können in bestimmten Fällen – etwa bei komplexen, nicht-linearen Strukturen oder hohen Anteilen an Ausreißern – Alternativen wie **Spectral Clustering**, **Agglomerative Clustering**, **Gaussian Mixture Models** oder **Bayesian Gaussian Mixture Models** überlegen sein. Dabei erweist sich insbesondere das BGMM als leistungsstark, wenn die wahre Anzahl der Cluster nicht bekannt ist oder Unsicherheiten über Daten und Modellierung bestehen; allerdings sind solche Verfahren meist rechenintensiver und verlangen eine sorgfältige Parametrisierung.

Die vergleichsweise geringe Zahl von *reinen Clustering-Wettbewerben* auf Kaggle verdeutlicht, dass das unüberwachte Lernen im Vergleich zu überwachten Methoden (z. B. Klassifikation oder Regression) noch immer eine eher untergeordnete Rolle spielt. Gerade für industrielle und forschungsnahe Anwendungen bietet das Clustering jedoch vielseitige Einsatzmöglichkeiten – etwa in der Segmentierung von Kundenprofilen oder der Erkennung verborgener Muster in hochdimensionalen Datensätzen. Die *zentrale Herausforderung* besteht dabei stets in der *abstimmten Auswahl und Feinjustierung* des Algorithmus, um die spezifischen Charakteristika eines Datensatzes optimal zu erfassen.

IX. Ausblick

Die Ergebnisse dieser Arbeit eröffnen mehrere Möglichkeiten für zukünftige Untersuchungen im Bereich des unüberwachten Lernens und der Clustering-Algorithmen. Für weiterführende Studien empfehlen sich insbesondere folgende Ansätze:

1) **Erweiterung der Datenquellen:** Die Analyse weiterer Plattformen wie GitHub oder anderer wissenschaftlicher Archive könnte ein vollständigeres Bild über die Nutzung und Weiterentwicklung von Clustering-Algorithmen bieten.
2) **Umfassende Notebook-Analyse:** Anstelle der ausschließlichen Betrachtung von Wettbewerbsbeiträgen wäre es sinnvoll, alle öffentlich zugänglichen Notebooks zu untersuchen, in denen Clustering-Methoden verwendet werden. Dabei sollte erfasst werden, welche Datensätze genutzt werden und welche Faktoren zu einer besseren Leistung der Algorithmen führen.
3) **Aktuelle Wettbewerbsanalysen:** Die Untersuchung aller derzeit aktiven Wettbewerbe auf Plattformen wie Kaggle kann helfen, aktuelle Trends und Vorlieben der Community bezüglich der eingesetzten Clustering-Methoden zu erkennen. Dies unterstützt das Verständnis der Entwicklungen im unüberwachten Lernen und hilft, zukünftige Forschungsschwerpunkte zu definieren.

Zusammengefasst zeigt diese Studie, dass die Auswahl des passenden Clustering-Algorithmus immer von den spezifischen Daten und dem jeweiligen Anwendungsfall abhängt. Die vorgeschlagenen Erweiterungen bieten einen guten Rahmen, um noch besser zu verstehen, welche Faktoren den Erfolg von Clustering-Algorithmen beeinflussen. Dies ist ein wichtiger Schritt, um in praktischen Anwendungen und in der zukünftigen Forschung noch effektivere Lösungen zu entwickeln.

Literatur

[1] R. Xu und D. Wunsch, “Survey of Clustering Algorithms”, *IEEE Transactions on Neural Networks*, Jg. 16, Nr. 3, S. 645–678, 2005.
[2] A. K. Jain, M. N. Murty und P. J. Flynn, “Data clustering: A review”, *ACM Computing Surveys*, 1999.
[3] A. Y. Ng, M. I. Jordan und Y. Weiss, “On spectral clustering: Analysis and an algorithm”, *Advances in neural information processing systems*, Jg. 14, S. 849–856, 2002.
[4] L. Kaufman und P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990, ISBN: 978-0471878766.
[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[6] D. M. Blei und M. I. Jordan, “Variational inference for Dirichlet process mixtures”, *Bayesian Analysis*, Jg. 1, Nr. 1, S. 121–143, 2006.
[7] M. Ester, H.-P. Kriegel, J. Sander und X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD’96)*, 1996, S. 226–231.
[8] C. Fraley und A. E. Raftery, “Model-based clustering, discriminant analysis, and density estimation”, *Journal of the American Statistical Association*, Jg. 97, Nr. 458, S. 611–631, 2002.
[9] P. J. Rousseeuw, “Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis”, *Journal of Computational and Applied Mathematics*, Jg. 20, S. 53–65, 1987.
[10] L. Hubert und P. Arabie, “Comparing partitions”, *Journal of Classification*, Jg. 2, Nr. 1, S. 193–218, 1985.
[11] Çağlar Uslu. (2022). What is Kaggle? Zugriff am 5. Februar 2025, Adresse: https://www.datacamp.com/blog/what-is-kaggle.
[12] ——, (2022). Kaggle Competitions: The Complete Guide. Zugriff am 5. Februar 2025, Adresse: https://www.datacamp.com/blog/kaggle-competitions-the-complete-guide.

[13] R. V. Krejcie und D. W. Morgan, “Determining sample size for research activities”, *Educational and Psychological Measurement*, Jg. 30, Nr. 3, S. 607–610, 1970.

[14] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

[15] D. J. Biau, S. Kernéïs und R. Porcher, “Statistics in brief: the importance of sample size in the planning and interpretation of medical research”, *Clinical Orthopaedics and Related Research*, Jg. 466, Nr. 9, S. 2282–2288, 2008.

[16] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman und A. Y. Wu, “An Efficient k-Means Clustering Algorithm: Analysis and Implementation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 24, S. 881–892, 2002. DOI: 10.1109/TPAMI.2002.1017616.

[17] U. von Luxburg, “A Tutorial on Spectral Clustering”, *Statistics and Computing*, Jg. 17, S. 395–416, 2007. DOI: 10.1007/s11222-007-9033-z.

[18] F. Murtagh, “Algorithms for Hierarchical Clustering: An Overview”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Jg. 2, S. 86–97, 2012. DOI: 10.1002/widm.53.

[19] X. Lin, X. Yang und Y. Li, “A Deep Clustering Algorithm based on Gaussian Mixture Model”, in *Journal of Physics: Conference Series*, Bd. 1302, 2019, S. 032 012. DOI: 10.1088/1742-6596/1302/3/032012.

[20] J. Lu, “A survey on Bayesian inference for Gaussian mixture model”, *arXiv preprint arXiv:2108.11753*, 2021. Adresse: https://arxiv.org/abs/2108.11753.