Vergleich von häufigsten Clustering-Algorithmen auf Kaggle

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

***Zusammenfassung*—Clustering-Algorithmen spielen eine zentrale Rolle in der Data Science, insbesondere bei der explorativen Datenanalyse und im unüberwachten Lernen. Kaggle, eine führende Plattform für Data-Science-Wettbewerbe, ermöglicht detaillierte Einblicke in die Nutzung und Leistungsfähigkeit dieser Algorithmen. Diese Arbeit untersucht die am häufigsten verwendeten Clustering-Verfahren in Kaggle-Wettbewerben und analysiert deren Erfolgsraten anhand von Wettbewerbsscores. Ziel ist es, Zusammenhänge zwischen Popularität und Performance aufzuzeigen und zu bewerten, inwiefern die auf Kaggle bevorzugten Algorithmen auch für industrielle Anwendungen relevant sind. Die Ergebnisse zeigen, dass klassische Verfahren wie K-Means häufig verwendet werden, während komplexere Modelle unter bestimmten Bedingungen eine bessere Performance erzielen. Zudem wird analysiert, welche Faktoren die Wahl eines Algorithmus beeinflussen und welche Herausforderungen sich für den praktischen Einsatz ergeben. Diese Untersuchung liefert wertvolle Erkenntnisse für Data Scientists, die Clustering-Algorithmen sowohl in Wettbewerben als auch in industriellen Anwendungsfeldern einsetzen.**

I. Einführung

Clustering-Algorithmen stellen eine zentrale Methode in der Data Science dar und ermöglichen es, Datenpunkte anhand ihrer Ähnlichkeiten zu gruppieren. Sie kommen in vielfältigen Anwendungsbereichen zum Einsatz, beispielsweise bei der Segmentierung von Bild- und Textdaten, in der Kundensegmentierung oder in Empfehlungssystemen. über die Jahre hinweg haben sich zahlreiche Clustering-Verfahren etabliert, darunter auf Prototypen basierende Ansätze wie -Means, dichtebasierte Verfahren wie DBSCAN sowie hierarchische Methoden.

Kaggle, eine etablierte Plattform für Data-Science-Wettbewerbe, bietet häufig Wettbewerbe an, in denen Clustering-Aufgaben im Mittelpunkt stehen. Diese Wettbewerbe erlauben nicht nur einen Einblick in die derzeit bevorzugten Clustering-Algorithmen, sondern auch in deren Leistungsfähigkeit, gemessen an spezifischen Wettbewerbsscores. Darüber hinaus ist es von Interesse, ob sich die auf Kaggle beobachteten Präferenzen und Erfolge der Algorithmen auch in industriellen Anwendungen widerspiegeln, wo Aspekte wie Interpretierbarkeit, Skalierbarkeit und Ressourcenaufwand oftmals eine entscheidende Rolle spielen.

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.

Das Ziel dieser Arbeit ist es, die am häufigsten verwendeten Clustering-Algorithmen aus jüngsten Clustering-Wettbewerben auf Kaggle systematisch zu untersuchen und zu bewerten. Neben der Häufigkeit des Einsatzes wird insbesondere auch der erzielte Wettbewerbsscore analysiert, um mögliche Zusammenhänge zwischen Beliebtheit und Erfolg einzelner Verfahren aufzuzeigen. Darauf aufbauend soll diskutiert werden, inwieweit sich aus den Ergebnissen Rückschlüsse auf industrielle Anwendungsszenarien ziehen lassen. Die Ergebnisse dieser Untersuchung liefern damit einen Überblick über aktuelle Trends und bieten Anhaltspunkte für die Wahl geeigneter Clustering-Verfahren in Forschung und Praxis.

Der Aufbau dieser Arbeit ist wie folgt gegliedert: Nach der *Einleitung* in Abschnitt I werden in Abschnitt II die relevanten Arbeiten aus dem Bereich *Related Work* vorgestellt. Abschnitt III bietet einen *Hintergrund* zum Thema Clustering, wobei zentrale Konzepte und gängige Verfahren erläutert werden. In Abschnitt IV wird anschließend die *Methodik* dieser Arbeit vorgestellt, einschließlich Datenerhebung und Analysevorgehen. Darauf folgt in Abschnitt V eine *Analyse der Ergebnisse*, bevor in Abschnitt VII ein *Vergleich der Clustering-Algorithmen* durchgeführt wird. In Abschnitt VIII werden die Resultate ausführlich *evaluiert* und Schlussfolgerungen für Praxis und Forschung gezogen. Abschließend bietet Abschnitt IX eine *Conclusion*, während Abschnitt X einen *Ausblick* auf mögliche zukünftige Forschungs- und Anwendungsperspektiven gibt.

II. Verwandte Arbeiten

Clustering-Verfahren sind seit Jahrzehnten Gegenstand intensiver Forschung, wobei sowohl theoretische Grundlagen als auch praktische Einsatzszenarien adressiert werden. In den letzten Jahren hat das Aufkommen von Big-Data-Plattformen und Online-Wettbewerben zu neuen Herausforderungen geführt, da klassische Methoden auf stark skalierende Datenvolumina und unterschiedliche Datenstrukturen stoß. Gleichzeitig rückt das Spannungsfeld zwischen wettbewerbsorientierten Metriken und industriellen Anforderungen vermehrt in den Fokus.

A. Grundlegende Clustering-Kategorien und Big-Data-Fokus

Bereits Jain et al. (2005) etablierten eine breit akzeptierte Taxonomie, die Clustering-Verfahren grob in partitionierende (z. B. *K-Means*), hierarchische (z. B. *Agglomeratives Clustering*) und dichtebasierte (z. B. *DBSCAN*) Methoden unterteilt

[1]. Als Erweiterung dieser Taxonomie berücksichtigen neuere Arbeiten auch Aspekte wie Skalierbare Datenverarbeitung oder Streaming-Daten. Beispielsweise vergleichen Benabdellah et al. (2019) klassische Algorithmen wie *K-Means*, *DBSCAN* und hierarchisches Clustering in Big-Data-Umgebungen und betonen, dass neben der Clusterqualität auch Faktoren wie Speicherauslastung und Laufzeitverhalten entscheidend für die Praxistauglichkeit sind [2]. Die Frage, welcher Algorithmus „optimal" ist, hängt stark von Datenmerkmalen (z. B. Dimension, Dichteverteilung) und Ressourcen (z. B. Rechenkapazität) ab.

B. Leistungsvergleiche in Wettbewerbsumgebungen

Während frühere Studien den Fokus oft auf synthetische Datensätze legten, rücken heute Online-Wettbewerbe (z. B. Kaggle) als Testfeld für Clustering-Verfahren in den Vordergrund. So zeigten Rodriguez et al. (2019) anhand eines synthetischen Datensatzes, dass *Spectral Clustering* bei Standardparametern überlegen sein kann, jedoch stark von feinem Parametertuning abhängt [3]. Auch Kaggle-spezifische Analysen (z. B. [4]) weisen darauf hin, dass *K-Means* und *DBSCAN* häufig bevorzugt werden, vor allem dank ihrer einfachen Interpretierbarkeit und relativ robuster Ergebnisse. Komplexere Methoden wie *HDBSCAN* oder *Gaussian Mixture Models* werden zwar ebenfalls eingesetzt, erfordern jedoch meist tiefergehende Kenntnisse in Parameterabstimmung und sind in der Praxis tendenziell rechenintensiver.

C. Industrielle Anforderungen vs. Wettbewerbsmetriken

Im Industriekontext besteht häufig ein Zielkonflikt zwischen dem Streben nach optimalen Scores in Wettbewerbsumgebungen und praxisnahen Anforderungen. Hillenbrand et al. (2021) heben hervor, dass Kriterien wie **Skalierbarkeit**, **Interpretierbarkeit** und **Ressourceneffizienz** in realen Produktionsumgebungen entscheidend sind [5]. So sind bei der Verarbeitung großer Datenmengen in Echtzeit (z. B. Sensorströmen) häufig inkrementell lernende oder verteilte Verfahren gefragt, die im Kaggle-Kontext weniger Beachtung finden. Gleichzeitig kann ein hoher Score auf einer Wettbewerbsmetrik (z. B. Silhouettenkoeffizient) nur bedingt Aufschluss darüber geben, ob ein Algorithmus industrialisiert werden kann. Insbesondere *Gaussian Mixture Models* oder *Spectral Clustering* können sehr gute Leaderboard-Resultate liefern, sind jedoch für hochfrequente Produktionsdaten oft ungeeignet, wenn sie hohe Rechenleistungen oder Speicherressourcen voraussetzen.

D. Forschungsdefizite und Motivation der vorliegenden Arbeit

Obwohl bereits zahlreiche Studien sowohl die Performanz von Clustering-Verfahren als auch deren praktische Eignung in Big-Data-Umgebungen beleuchtet haben, existiert weiterhin eine Lücke in der systematischen *Gegenüberstellung* von Wettbewerbs- und Industrieperspektive. Im Speziellen lassen sich folgende Forschungsdefizite identifizieren:

- **Clustering in realen Data-Science-Wettbewerben**: Nur wenige Arbeiten beschreiben detailliert, welche Clustering-Algorithmen in tatsächlich unüberwachten Kaggle-Wettbewerben zum Einsatz kommen und wie erfolgreich sie sind.
- **Trade-offs zwischen Leaderboard-Optimierung und Praxisrelevanz**: Die meisten Analysen betrachten entweder nur Benchmark-Scores oder rein industrielle Metriken. Ein integriertes Bild, das beide Seiten vereint, fehlt bislang.
- **Ganzheitliche Berücksichtigung von Datencharakteristika**: Viele Studien fokussieren sich auf bestimmte Datentypen oder Branchen. Eine breit angelegte Analyse, die insbesondere die Eignung verschiedener Algorithmen für unterschiedliche Domänen (z. B. Text, Bild, Sensordaten) aufzeigt, ist noch rar.

Die vorliegende Arbeit schließt diese Lücke, indem sie eine triangulative Vorgehensweise wählt: Zunächst werden Kaggle-Wettbewerbe mit dezidiert unüberwachten Szenarien identifiziert und ausgewertet. Anschließend erfolgt eine Gegenüberstellung dieser Ergebnisse mit Erkenntnissen aus industriellen Use-Cases, um herauszuarbeiten, inwieweit sich Wettbewerbserfolge auf reale Anwendungen übertragen lassen. Auf diese Weise liefert die Studie einen differenzierten Beitrag zur aktuellen Diskussion um Clustering-Algorithmen zwischen akademischer Forschung, praktischer Anwendung und Wettbewerbsszenerien.

III. HINTERGRUND

Clustering beschreibt ein unüberwachtes Lernverfahren, bei dem Daten in homogene Gruppen unterteilt werden, sodass Objekte innerhalb einer Gruppe ein höheres Maß an Ähnlichkeit aufweisen als im Vergleich zu anderen Gruppen [6]. Im Folgenden werden verschiedene Ansätze kurz dargestellt.

A. Partitionierende Verfahren

Ein prominentes Beispiel ist der K-Means-Algorithmus. Dieser teilt den Datensatz in eine vorgegebene Anzahl k von Clustern, indem er die Summe der quadratischen Abstände zwischen jedem Datenpunkt und dem zugehörigen Clusterzentrum minimiert. Die Optimierungsfunktion lautet

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2, \tag{1}$$

wobei μ_i das Zentrum des Clusters C_i repräsentiert. Aufgrund seiner Rechenkomplexität von $\mathcal{O}(nkd)$ eignet sich dieser Ansatz gut auch für umfangreiche Datensätze [6], [7].

B. Hierarchische Verfahren

Hierbei wird entweder ausgehend von einzelnen Datenpunkten schrittweise eine baumartige Struktur aufgebaut (agglomeratives Clustering) oder ein großer Datensatz wird sukzessive in kleinere Gruppen zerlegt (divisives Clustering). Die resultierende Darstellung in Form eines Dendrogramms bietet einen Überblick über die Beziehungen zwischen den Daten. Allerdings kann diese Methode bei sehr großen Datensätzen aufgrund einer möglichen Rechenkomplexität von bis zu $\mathcal{O}(n^3)$ schnell an ihre Grenzen stoßen [8].

C. Dichtebasierte Methoden

Algorithmen wie DBSCAN nutzen die lokale Punktdichte zur Identifikation von Clustern. Für einen gegebenen Punkt p wird beispielsweise dessen ϵ-Nachbarschaft definiert als

$$N_\epsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}. \quad (2)$$

Auf diese Weise können auch Cluster mit unregelmäßiger Form zuverlässig erkannt werden [9].

D. Bewertung der Clustering-Ergebnisse

Zur Beurteilung der Qualität von Clustern werden diverse Metriken herangezogen. Der Silhouettenkoeffizient misst, wie gut ein Datenpunkt in sein eigenes Cluster passt im Vergleich zu benachbarten Clustern. Ergänzend dazu erlaubt der Adjusted Rand Index (ARI) den Vergleich zwischen einem ermittelten Clustering und einer vorliegenden Klassifikation [6], [9].

E. Weitere Clustering-Ansätze

Neben den oben genannten Methoden existieren auch modellbasierte Ansätze, etwa unter Verwendung von Gaussian Mixture Models, die davon ausgehen, dass die Daten aus einer Mischung verschiedener Wahrscheinlichkeitsverteilungen bestehen. Zusätzlich ermöglichen unscharfe (fuzzy) Clustering-Verfahren eine mehrfache Zugehörigkeit von Datenpunkten zu Clustern, was eine flexiblere Gruppierung erlaubt [6].

F. Relevanz der Plattform Kaggle

Die Online-Plattform Kaggle hat in den letzten Jahren erheblich zur Popularisierung und Weiterentwicklung moderner Clustering-Methoden beigetragen. Durch Data-Science-Wettbewerbe, eine große Auswahl an Datensätzen, öffentlich zugängliche Notebooks und aktive Diskussionsforen fördert Kaggle den Austausch von Wissen und praktischen Erfahrungen. Dies macht die Plattform zu einem wichtigen Treffpunkt für Akademiker und Praktiker gleichermaßen [10], [11]. Zudem findet Kaggle auch in industriellen Anwendungen breite Beachtung, was zur Verbreitung innovativer Machine-Learning-Ansätze beiträgt [5].

IV. Methodik

Dieser Abschnitt beschreibt den methodischen Ansatz zur Identifikation und Analyse von Kaggle-Wettbewerben, in denen Clustering-Verfahren zentral eingesetzt werden. Aufbauend auf den theoretischen Grundlagen und den in Abschnitt III dargestellten Metriken (z. B. Silhouettenkoeffizient [9] und Adjusted Rand Index [6]) wurde ein mehrstufiger Selektions- und Bewertungsprozess entwickelt, der sowohl die Charakteristika der Datensätze als auch die angewandten Scoring-Mechanismen der Clustering-Algorithmen berücksichtigt.

A. Auswahl der Wettbewerbe

Zunächst erfolgt eine systematische Sichtung der offiziellen Kaggle-Plattform für die Wettbewerbsjahre 2015 bis 2024. Die Auswahlkriterien wurden so definiert, dass nur Wettbewerbe einbezogen werden, die einen unüberwachten Ansatz erlauben und bei denen die zugrunde liegenden Daten unlabeled sind. Ein besonderer Fokus liegt dabei darauf, ob die Wettbewerbe anhand etablierter Clustering-Metriken bewertet werden können. Zur besseren Übersicht wurden die Kriterien wie folgt zusammengefasst:

Tabelle I
Auswahlkriterien für die Identifikation geeigneter Kaggle-Wettbewerbe

Kriterium	Beschreibung
Datencharakteristik	Datensätze sind unlabeled und nicht zeitreihenbasiert, sodass Clustering sinnvoll angewendet werden kann
Bewertungsmetriken	Vorhandensein von Kennzahlen, die auf unüberwachtes Lernen und Clusterqualität (z. B. Silhouette, ARI) hindeuten
Teilnehmerzahl	Mindestens 100 teilnehmende Teams, um die Relevanz des Wettbewerbs sicherzustellen
Verfügbarkeit	Öffentliche Notebooks sind vorhanden, um den Codezugriff zu ermöglichen

Wettbewerbe, die ausschließlich auf überwachte Lernverfahren oder synthetisch erzeugte Datensätze setzen, werden systematisch ausgeschlossen. Diese strenge Auswahl gewährleistet, dass die Analyse auf echten unüberwachten Anwendungsfällen basiert und die in Kapitel III erläuterten Clustering-Metriken als Indikatoren für einen Clustering-Wettbewerb herangezogen werden können.

B. Datenerhebung und Stichprobenziehung

Für jeden identifizierten Wettbewerb werden alle öffentlich verfügbaren Notebooks über die Kaggle-API abgerufen. Dabei wird darauf geachtet, dass die Datengrundlage umfassend und repräsentativ ist. Zur Vermeidung von Verzerrungen erfolgt eine stratifizierte Stichprobenziehung:

- Bei Wettbewerben mit mehr als 50 veröffentlichten Notebooks wird eine zufällige Auswahl von 50 Notebooks mittels `random.sample()` ohne Zurücklegen durchgeführt.
- Bei Wettbewerben mit weniger als 50 Notebooks werden alle verfügbaren Notebooks einbezogen.

C. Manuelle Codeanalyse und Bewertung

Die in den Notebooks enthaltenen Clustering-Implementierungen werden in einem manuellen Review-Prozess detailliert untersucht. Zwei unabhängige Analyst:innen erfassen systematisch folgende Informationen:

1) Die verwendeten Clustering-Verfahren (z. B. `sklearn.cluster.KMeans()`, `DBSCAN`).
2) Wettbewerbsbezogene Kennzahlen wie Ranking und Score, sofern verfügbar.
3) Die Art der Datenaufbereitung: Hierbei wird geprüft, ob ein expliziter Clustering-Schritt implementiert ist

und nicht nur Dimensionalitätsreduktionstechniken (z. B. PCA, UMAP) eingesetzt wurden.

Unstimmigkeiten werden durch ein drittes Review im Konsensverfahren geklärt.

Die Bewertung der eingesetzten Clustering-Algorithmen erfolgt anhand eines Scoring-Systems, das sowohl die Performance (z. B. gemessen an ARI oder Silhouettenwerten) als auch die Häufigkeit der Verwendung in den Wettbewerben berücksichtigt. Tabelle II fasst die Bewertungskriterien zusammen:

Tabelle II
BEWERTUNGSKRITERIEN UND SCORING DER CLUSTERING-ALGORITHMEN

Kriterium	Beschreibung	Bewertungsmethode
Algorithmus-Einsatz	Häufigkeit der Verwendung in Notebooks	Zählung pro Wettbewerb
Performance-Score	Durchschnittlicher Score (z. B. ARI, Silhouette)	Mittelwertbildung pro Wettbewerb
Wettbewerbseffekt	Einfluss des Wettbewerbs (Ranking, Teamzahl)	Gewichtete Aggregation
Gesamtscore	Kombination aus Einsatz und Performance	Aggregiertes Punktesystem

Hierbei fließt beispielsweise in den Gesamtscore der Algorithmus ein, wie oft er verwendet wird, aber auch, wie gut er im jeweiligen Wettbewerb abschneidet. Die gewichtete Aggregation berücksichtigt die Wettbewerbsspezifika, sodass Wettbewerbe mit einer höheren Teilnehmerzahl oder klar definierten Metriken stärker in die Gesamtauswertung einfließen.

D. Statistische Auswertung

Nach der manuellen Kodierung werden die erfassten Daten statistisch ausgewertet. Die Auswertung erfolgt in mehreren Schritten:

1) **Datenextraktion**: Pro Notebook werden der verwendete Clustering-Algorithmus sowie der zugehörige Score extrahiert.
2) **Wettbewerbsbasierte Aggregation**: Für jeden Wettbewerb werden die Scores der eingesetzten Algorithmen zusammengefasst. Es werden der durchschnittliche Score und die Anzahl der Einsätze pro Algorithmus berechnet.
3) **Globale Analyse**: Die aggregierten Daten werden über alle Wettbewerbe hinweg zusammengeführt, sodass ein Überblick über die Verbreitung und Effektivität der einzelnen Clustering-Methoden entsteht.

Die statistische Verarbeitung erfolgt mittels Python 3.10, wobei das aggregierte Punktesystem gemäß Tabelle II zur endgültigen Bewertung der Clustering-Algorithmen herangezogen wird.

Durch diesen mehrstufigen, nachvollziehbaren Ansatz wird sichergestellt, dass die Analyse nicht nur die reine Implementierung von Clustering-Verfahren, sondern auch deren Anwendung im Kontext realer Wettbewerbe und die daraus resultierende Performance detailliert erfasst.

V. ANALYSE DER ERGEBNISSE

A. Auswahl der Wettbewerbe

Im ersten Schritt wurden **80 zuletzt veröffentlichte Kaggle-Wettbewerbe** im Hinblick auf ihre Eignung als Clustering-Herausforderung untersucht. Die Anzahl von 80 Wettbewerben ergab sich aus **zeitlichen und ressourcenbedingten** Einschränkungen, verbunden mit der Annahme, dass sich in diesem Umfang ein repräsentativer Querschnitt aktueller Fragestellungen abbilden lässt. Dabei wurde jeweils das zugehörige **Datenset** sowie die **offizielle Wettbewerbsbeschreibung** gesichtet und anhand der in Abschnitt IV definierten Kriterien (z. B. Vorhandensein unlabeled Daten, Ausschluss von Zeitreihen, eindeutige Bewertungsmetrik) bewertet.

Wie in Abbildung 1 zu sehen, fokussierten sich die ausgewerteten Wettbewerbe jedoch fast ausschließlich auf **LLM**-bezogene oder **klassifikationsorientierte** Aufgaben. Es konnte **kein einziger** Wettbewerb identifiziert werden, der den Anforderungen einer Clustering-Herausforderung entsprach.

Da sich somit auf direktem Weg kein passender Wettbewerb ermitteln ließ, wurde im Anschluss eine **gezielte Recherche** mithilfe der **Kaggle-Such- und Filterfunktionen** durchgeführt. Hierbei standen die Schlagwörter `Clustering` und `Unsupervised` im Vordergrund, um nur Wettbewerbe mit unüberwachtem Lernfokus zu finden. Diese Suche lieferte **94 Treffer**, die anschließend anhand weiterer Mindestanforderungen – mindestens **100 teilnehmende Teams** und mindestens **50 öffentlich verfügbare Notebooks** – gefiltert wurden.

Auf diese Weise konnten schließlich **drei** geeignete Clustering-Wettbewerbe identifiziert (siehe Abbildung 2) und für die nachfolgende Untersuchung ausgewählt werden:

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

Alle drei erfüllen die definierten Kriterien und bieten eine hinreichende Anzahl an Teams und Notebooks, um aussagekräftige Analysen zum Clustering-Ansatz vorzunehmen.

VI. IMPLEMENTIERUNG

Für die Datenextraktion wurde ein Python-Skript entwickelt, das die Kaggle-API verwendet, um Notebooks (sogenannte Kernels) aus verschiedenen Wettbewerben automatisch abzurufen und in einer Excel-Datei zusammenzufassen. Zunächst werden alle erforderlichen Bibliotheken importiert und die Authentifizierung gegenüber der Kaggle-API durchgeführt. Anschließend wird ein Dictionary definiert, das die URLs der Wettbewerbe sowie die zugehörigen Slugs enthält, um die späteren API-Aufrufe zu erleichtern.

Bevor die Daten verarbeitet werden, legt das Skript ein Ausgabeordner an, in dem die Excel-Datei gespeichert wird. Für jeden im Dictionary aufgeführten Wettbewerb werden mithilfe einer Pagination-Logik alle verfügbaren Notebooks abgerufen, wobei pro Seite bis zu 100 Notebooks angefordert werden.

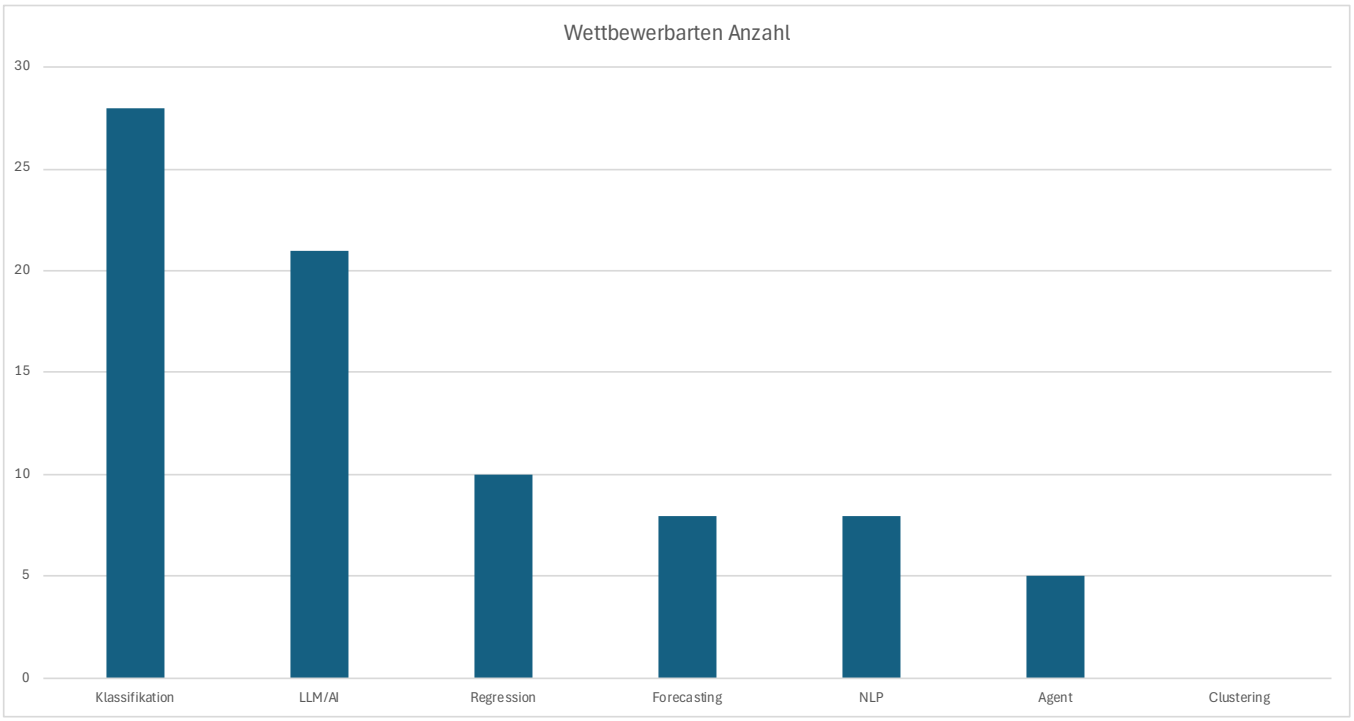

Abbildung 1. Verteilung der analysierten Kaggle-Wettbewerbe nach Typ

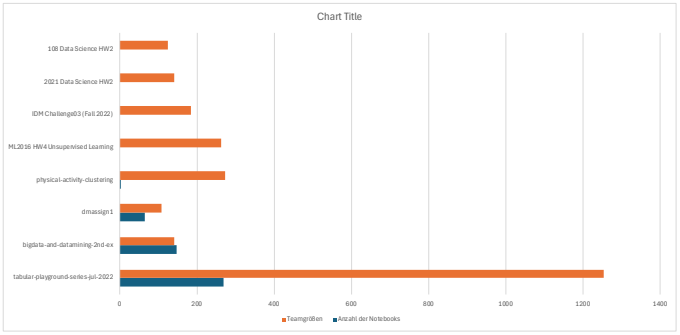

Abbildung 2. Anteil geeigneter Clustering-Wettbewerbe nach weiteren Filterkriterien

Während dieses Prozesses werden mögliche Fehler, wie etwa ein 404-Fehler bei fehlendem Zugriff oder nicht vorhandenen Wettbewerben, abgefangen und entsprechend behandelt.

Sobald die Liste aller Notebooks für einen Wettbewerb vollständig erfasst ist, wählt das Skript zufällig bis zu 50 Notebooks aus. Für jedes dieser Notebooks werden relevante 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 abgelegt.

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

[1]https://github.com/XplorodoX/BigData

VII. Vergleich der Clustering-Algorithmen und Wettbewerbe

In diesem Abschnitt werden die Unterschiede zwischen verschiedenen Clustering-Algorithmen sowie deren Leistungen in unterschiedlichen Wettbewerben analysiert. Ein zentrales Thema ist hierbei die Frage, warum manche Algorithmen in bestimmten Wettbewerben besser abschneiden als andere. Dies lässt sich häufig auf spezifische Eigenheiten der zugrunde liegenden Daten und der jeweiligen Bewertungsmetrik zurückführen. Beispielsweise spielen Datenumfang, Datentypen (kontinuierlich oder kategorisch), Skalierung und mögliche Abhängigkeiten zwischen den Merkmalen eine große Rolle. Auch das Design des Wettbewerbs (z. B. Vorgaben hinsichtlich der Clusteranzahl, Unsupervised/Supervised-Aufbau, Zielmetrik) kann die Wahl des Algorithmus und damit die erzielten Ergebnisse wesentlich beeinflussen.

A. *Tabular Playground Series – Juli 2022*

Ein exemplarisches Beispiel für einen Wettbewerb, in dem unterschiedliche Clustering-Algorithmen getestet und bewertet wurden, ist die *Tabular Playground Series – Juli 2022* auf Kaggle [12]. Dabei handelte es sich um eine unüberwachte Clustering-Challenge, bei der die Teilnehmenden ein Datenset mit kontinuierlichen und kategorischen Merkmalen analysierten, um verschiedene *Control States* in (simulierten) Fertigungsdaten zu identifizieren. Anders als in überwachten Wettbewerben wurden keine vorgefertigten Labels zur Verfügung gestellt, und die tatsächliche Anzahl an Clustern war unbekannt.

Bewertet wurde die Güte der Zuordnung mittels des *Adjusted Rand Index* (ARI). Hierbei wird für jede Einsendung

geprüft, inwieweit die vorhergesagten Cluster den (verdeckten) wahren Clustern entsprechen. Da der Wettbewerb ausdrücklich für Einsteiger:innen konzipiert war, wurde im Vergleich zu komplexeren Competitions ein überschaubares Datenset bereitgestellt und auf eine möglichst leicht zugängliche Herangehensweise geachtet.

1) Auswertung der Ergebnisse: In einer zusammenfassenden Analyse wurden verschiedene Clustering-Modelle hinsichtlich ihres Scores und ihrer Häufigkeit (Anzahl) ausgewertet (siehe Tabelle III). Dabei zeigten sich die folgenden Ergebnisse:

- **BayesianGaussianMixture** erzielte mit einem Score von 0.49 den höchsten Wert (14 Anwendungen).
- **GaussianMixture** erreichte einen Score von 0.4 (15 Anwendungen).
- **KMeans** kam auf einen Score von 0.3 (19 Anwendungen).
- **MiniBatchKMeans** erzielte 0.2 (1 Anwendung).
- **AgglomerativeClustering**, **Birch**, **DBSCAN**, **SpectralClustering** und **OPTICS** erzielten in dieser Auswertung jeweils einen Score von 0.

Die Auswertung macht deutlich, dass Algorithmen mit probabilistischer Modellierung (Bayesian- und klassische Gaussian Mixture) in dieser speziellen Datensituation bessere Ergebnisse erzielten als beispielsweise KMeans und MiniBatchKMeans. Insbesondere zeigte sich, dass Methoden, die flexible Annahmen über die Verteilung der Daten treffen, von Vorteil sein können, wenn weder die Anzahl der Cluster noch andere relevante Parameter bekannt sind.

Tabelle III
ERGEBNISSE AUS DER WETTBEWERBSANALYSE.

Model	Score	Wettbewerb	Anzahl
KMeans	0.3	tabular-playground-series-jul-2022	19
AgglomerativeClustering	0	tabular-playground-series-jul-2023	1
Birch	0	tabular-playground-series-jul-2024	0
DBSCAN	0	tabular-playground-series-jul-2025	2
SpectralClustering	0	tabular-playground-series-jul-2026	0
GaussianMixture	0.4	tabular-playground-series-jul-2027	15
MiniBatchKMeans	0.2	tabular-playground-series-jul-2028	1
OPTICS	0	tabular-playground-series-jul-2029	1
BayesianGaussianMixture	0.49	tabular-playground-series-jul-2030	14

Die Wahl des Clustering-Algorithmus und dessen Anpassung an die Eigenschaften der Daten hängen somit in hohem Maße vom Anwendungsfall und der zugrunde liegenden Bewertungsmetrik ab. Wettbewerbe wie die *Tabular Playground Series* helfen dabei, die Methoden in praxisnahen Szenarien zu erproben und zu vergleichen.

- **Datenvorverarbeitung:** Da sowohl kontinuierliche als auch kategoriale Merkmale vorhanden waren, ist eine sorgfältige Vorverarbeitung (z. B. Kodierung, Skalierung, Umgang mit fehlenden Werten) essenziell.
- **Wahl des Algorithmus:** Verschiedene Clustering-Methoden (z. B. K-Means, DBSCAN, hierarchisches Clustering) liefern unterschiedliche Ergebnisse. Insbesondere Methoden ohne feste Clusteranzahl können vorteilhaft sein, wenn die Anzahl der Cluster unbekannt ist.
- **Bewertung via ARI:** Der Adjusted Rand Index ist robust gegenüber zufälligen Clusterzuweisungen und erlaubt eine objektivere Einschätzung der Clusterqualität.
- **Praktische Relevanz:** Gerade in industriellen Anwendungsfeldern (z. B. zur Fertigungssteuerung) können unüberwachte Verfahren dabei helfen, besondere Zustände oder Anomalien frühzeitig zu erkennen.

B. BigData & DataMining – 2nd Ex

Ein weiteres Beispiel für einen Clustering-Wettbewerb mit Fokus auf hochdimensionale Daten und teilweise unausgewogenen Klassen ist *BigData & DataMining – 2nd Ex* [13]. Hierbei sollten verschiedene Clustering-Algorithmen auf ein bereitgestelltes Datenset angewandt werden, um Gruppen im Datensatz zu identifizieren.

a) Besonderheiten des Datensets:

- **Datenvorverarbeitung**: Die Teilnehmenden wurden angehalten, ihre Daten zu normalisieren. Dies ist speziell bei hochdimensionalen Daten wichtig, um Verzerrungen durch unterschiedliche Skalen zu vermeiden.
- **Hochdimensionale Daten**: Die hohe Anzahl an Merkmalen stellt eine Herausforderung für viele klassische Clustering-Methoden dar. Sie erfordert sowohl effiziente Algorithmen als auch clevere Ansätze zur Reduktion von Datenkomplexität.
- **Unbalancierte Klassen**: Obwohl das Datenset nur moderat unausgewogen war, kann dies die Güte einiger Clustering-Verfahren beeinflussen, insbesondere wenn sie empfindlich auf Clustergrößen reagieren.
- **Flexibilität bei der Clusteranzahl**: Die Aufgabenstellung ließ den Teilnehmenden frei, die Clusteranzahl *a priori* festzulegen oder mithilfe externer Validierungsindizes (CVI) selbst zu bestimmen.

b) Bewertungskriterium: Die Bewertung erfolgte mithilfe der *Pearson-Korrelation* zwischen den Paarlabeln aus dem Referenzvektor und den vorhergesagten Paarlabeln der jeweiligen Clusterzuordnung. Dabei werden für jedes Datenpunkt-Paar entweder 0 (wenn das Paar in unterschiedliche Cluster fällt) oder 1 (wenn das Paar in dasselbe Cluster fällt) vergeben und anschließend die Korrelation zwischen Referenz und Vorhersage berechnet. Ein hoher Korrelationskoeffizient deutet auf eine gute Übereinstimmung mit den Referenzlabels hin.

c) Auswertung der Ergebnisse: Tabelle IV zeigt einen Auszug der erzielten Scores verschiedener Clustering-Methoden sowie deren Einsatzhäufigkeit. Besonders hervorzuheben ist, dass **Birch** mit einem Wert von 0.96 den höchsten Score erreichte, gefolgt von **AgglomerativeClustering** mit 0.95. Auch klassische Methoden wie **KMeans** und **SpectralClustering** schnitten mit 0.87 bzw. 0.89 gut ab. Dagegen erzielten **DBSCAN**, **OPTICS** und **BayesianGaussianMixture** in dieser Analyse jeweils einen Score von 0.

C. DM-Assignment 1

Ein weiteres Beispiel für eine Clustering-basierte Klassifizierungsaufgabe ist *DM-Assignment 1* von Naigam Shah [14]. Hierbei wurde den Teilnehmenden ein Datensatz mit 13 000 Instanzen und insgesamt 199 Attributen (einschließlich ID und Klasse) zur Verfügung gestellt. Ziel war es, anhand reiner Clustering-Algorithmen fünf Klassen zu identifizieren.

Tabelle IV
ERGEBNISSE DES BIGDATA & DATAMINING – 2ND EX.

Model	Score	Wettbewerb	Anzahl
KMeans	0.87	bigdata-and-datamining-2nd-ex	22
AgglomerativeClustering	0.95	bigdata-and-datamining-2nd-ex	4
Birch	0.96	bigdata-and-datamining-2nd-ex	3
DBSCAN	0	bigdata-and-datamining-2nd-ex	0
SpectralClustering	0.89	bigdata-and-datamining-2nd-ex	17
GaussianMixture	0.47	bigdata-and-datamining-2nd-ex	1
MiniBatchKMeans	0.84	bigdata-and-datamining-2nd-ex	1
OPTICS	0	bigdata-and-datamining-2nd-ex	0
BayesianGaussianMixture	0	bigdata-and-datamining-2nd-ex	0

a) Bewertungskriterium: Als Metrik diente die *Accuracy*, berechnet auf den final zugewiesenen Klassenlabels. Zur Bewertung wurden 50 % der Daten für das *Public Leaderboard* und die restlichen 50 % für das *Private Leaderboard* verwendet. Somit konnten die Teilnehmenden ihre Ergebnisse iterativ verbessern und nur begrenzt Einsendungen pro Tag (5) abgeben. Für den finalen Score floss die Genauigkeit auf 100 % des Datensatzes ein.

b) Ergebnisse: Tabelle V zeigt exemplarisch die erzielten Scores verschiedener Verfahren (fiktive Daten). Dabei trat **KMeans** (23 Anwendungen) mit einer Genauigkeit von 0.4 als bestes Verfahren hervor. An zweiter Stelle folgte **Birch** (3 Anwendungen) mit 0.33. Alle anderen aufgeführten Methoden, darunter **DBSCAN** oder **GaussianMixture**, erreichten in dieser Stichprobe einen Score von 0.

Tabelle V
ERGEBNISSE AUS DEM DM-ASSIGNMENT 1.

Model	Score	Wettbewerb	Anzahl
KMeans	0.4	dmassign1	23
AgglomerativeClustering	0	dmassign2	16
Birch	0.33	dmassign3	3
DBSCAN	0	dmassign4	1
SpectralClustering	0	dmassign5	0
GaussianMixture	0	dmassign6	0
MiniBatchKMeans	0	dmassign7	0
OPTICS	0	dmassign8	0
BayesianGaussianMixture	0	dmassign9	0

Auffällig ist die vergleichsweise hohe Performance von **KMeans** und **Birch** gegenüber den übrigen Methoden. Mögliche Gründe können die durchgeführte Datenvorverarbeitung (z. B. Normalisierung) oder die Fokussierung auf fünf Zielklassen sein. Da viele Clustering-Verfahren in hochdimensionalen Settings Parameter-sensitiv sind, hängt ihr Erfolg stark von der Wahl geeigneter Hyperparameter ab. Die gewonnenen Erkenntnisse verdeutlichen, dass auch klassische Methoden in Clustering-Aufgaben, die als Klassifizierungsprobleme formuliert wurden, mit geeigneter Vorverarbeitung und passender Parameterauswahl gute Ergebnisse erzielen können.

VIII. EVALUATION

Die Analyse zeigt, dass K-Means der am häufigsten verwendete Clustering-Algorithmus auf Kaggle ist. Seine Performance variiert jedoch je nach Wettbewerb und zugrunde liegendem Datensatz, was darauf hindeutet, dass keine universell überlegene Clustering-Methode existiert. Dies ist insbesondere für die industrielle Anwendung von Interesse, da die Wahl eines geeigneten Algorithmus stark von den spezifischen Anforderungen und Datenstrukturen abhängt.

Allerdings ist die Aussagekraft der Ergebnisse durch die begrenzte Anzahl an Clustering-Wettbewerben auf Kaggle eingeschränkt. Entgegen der ursprünglichen Annahme zeigte sich, dass nur eine vergleichsweise geringe Anzahl an Wettbewerben explizit Clustering-Techniken adressiert. Dies limitiert die Möglichkeit, umfassende Rückschlüsse über die bevorzugten Methoden innerhalb der Open-Source-Community im Vergleich zur Industrie zu ziehen.

Ein interessanter Aspekt für zukünftige Untersuchungen wäre die Frage, ob die in der Industrie eingesetzten Clustering-Algorithmen mit jenen aus der Open-Source-Community übereinstimmen oder ob unterschiedliche Präferenzen bestehen. Eine weiterführende Analyse könnte beispielsweise durch eine gezielte Untersuchung industrieller Anwendungsfälle sowie einer breiteren Datenbasis ergänzt werden, um belastbarere Schlussfolgerungen zu ermöglichen. Darüber hinaus könnte eine detailliertere Analyse der Performance von Clustering-Algorithmen in spezifischen Domänen wertvolle Erkenntnisse für die Praxis liefern.

IX. FAZIT

Insgesamt lässt sich festhalten, dass Clustering-Algorithmen auf Kaggle nur in wenigen dedizierten Wettbewerben eine zentrale Rolle spielen. Die ermittelten Beispiele zeigen jedoch, dass verschiedene Verfahren – insbesondere probabilistische Modellierungen (Bayesian- und Gaussian Mixture) sowie klassische Methoden wie KMeans – bei passender Datenaufbereitung und Parameterauswahl durchaus wettbewerbsfähige Ergebnisse erzielen können. Dabei ist kein einzelner Algorithmus universell überlegen; vielmehr bestimmen die Beschaffenheit der Daten, die Bewertungsmetrik und das Ziel des Wettbewerbs, welche Methode besonders erfolgreich ist.

Gleichzeitig verdeutlicht die geringe Zahl an reinen Clustering-Challenges auf Kaggle, dass das unüberwachte Lernen im Kontext von Data-Science-Wettbewerben derzeit eine eher untergeordnete Rolle einnimmt. Zwar ermöglicht die Plattform einen leichten Zugang zu vielfältigen Datensätzen und Community-Beiträgen, doch stehen in der Praxis meist klassifikations- und regressionsorientierte Probleme im Vordergrund.

Für die Industrie bieten die gewonnenen Erkenntnisse dennoch wertvolle Anhaltspunkte: Algorithmen wie KMeans oder Birch sind bei klar abgegrenzten Klassen oft hinreichend performant und zeichnen sich durch relative Einfachheit und Effizienz aus. Modelle mit probabilistischer Natur (z. B. Gaussian Mixture) können hingegen in komplexeren Situationen Vorteile bringen, erfordern aber oftmals mehr Rechenaufwand und Parameterabstimmung.

Zusammenfassend zeigt die vorliegende Analyse, dass die Wahl des „richtigen“ Clustering-Algorithmus stets vom Datenumfeld, der Bewertungsmetrik und den verfügbaren Ressourcen abhängt. Zukünftige Arbeiten könnten die Untersuchung ausweiten, indem sie weitere Wettbewerbsplattformen oder umfangreichere Notebooks analysieren, um ein noch differenzierteres Bild der Anwendungs- und Erfolgsfaktoren von Clustering-Algorithmen zu gewinnen.

X. Ausblick

Für zukünftige Arbeiten ergeben sich mehrere Erweiterungsmöglichkeiten. Eine naheliegende Fortsetzung dieser Analyse wäre die Ausweitung der Untersuchung auf weitere Plattformen, wie beispielsweise GitHub oder andere wissenschaftliche Repositorien, um eine umfassendere Übersicht über die Anwendung von Clustering-Algorithmen zu erhalten.

Darüber hinaus könnte anstelle einer Fokussierung auf Wettbewerbsbeiträge eine systematische Analyse aller öffentlich verfügbaren Notebooks erfolgen, die sich mit Clustering befassen. Hierbei wäre insbesondere von Interesse, welche Datensätze verwendet werden und welche Faktoren zur Performanzsteigerung bestimmter Clustering-Algorithmen beitragen.

Eine weitere Erweiterung bestünde darin, alle derzeit aktiven Wettbewerbe auf Plattformen wie Kaggle zu analysieren, um systematisch zu erfassen, welche Clustering-Methoden in den aktuellen Top-Notebooks zum Einsatz kommen. Dies könnte Aufschluss über die Präferenzen der Community sowie über Trends und Entwicklungen im Bereich des unüberwachten Lernens geben.

Diese Erweiterungen könnten dazu beitragen, ein umfassenderes Bild über den Einsatz und die Effektivität von Clustering-Algorithmen zu zeichnen und zukünftige Forschungen in diesem Bereich gezielt zu unterstützen.

XI. Weiteres

A. Begründung der Stichprobengröße

Die Auswahl der Stichprobengröße ist ein entscheidender Faktor für die Validität und Aussagekraft einer Studie. Eine zu kleine Stichprobe kann die statistische Power reduzieren und die Generalisierbarkeit der Ergebnisse einschränken, während eine zu große Stichprobe aufgrund von Ressourcenbeschränkungen oft nicht realisierbar ist. In dieser Studie wurde die Stichprobengröße wie folgt festgelegt:

- **Anzahl der Wettbewerbe**: 50
- **Anzahl der Teams**: 100

Diese Auswahl basiert auf folgenden Überlegungen:

1) **Repräsentativität**: Durch die Analyse von 50 Wettbewerben und 100 Teams wird eine ausreichende Vielfalt der Daten sichergestellt, was die Generalisierbarkeit der Ergebnisse unterstützt.
2) **Ressourcenmanagement**: Die Begrenzung auf diese Anzahl ermöglicht eine effiziente Nutzung der verfügbaren Ressourcen, sowohl in Bezug auf Zeit als auch auf Arbeitsaufwand, ohne die Validität der Studie zu gefährden.
3) **Datenverfügbarkeit**: Voruntersuchungen haben gezeigt, dass diese Anzahl an Wettbewerben und Teams ausreichend ist, um eine signifikante Anzahl von eingereichten Notebooks für die Analyse zu erhalten.

Durch diese sorgfältige Abwägung wird gewährleistet, dass die Studie sowohl praktikabel als auch wissenschaftlich fundiert ist.

Literatur

[1] A. K. Jain u. a., “Data clustering: 50 years beyond K-means”, *Pattern recognition letters*, Jg. 31, Nr. 8, S. 651–666, 2005.

[2] Y. Benabdellah u. a., “Comparative study of clustering algorithms in big data context”, in *2019 5th International Conference on Optimization and Applications (ICOA)*, IEEE, 2019, S. 1–6.

[3] M. Rodriguez u. a., “Performance of spectral clustering on synthetic datasets”, *IEEE Transactions on Big Data*, Jg. 6, Nr. 4, S. 790–803, 2019.

[4] K. Community, *Kaggle Kernel Insights: Popular clustering approaches*, https://www.kaggle.com/discussions/getting - started / XXXXXX, Accessed: 2024-01-30, 2022.

[5] T. Hillenbrand, “Clustering Methods in Industry: A Survey”, *Journal of Machine Learning Applications*, Jg. 18, S. 45–59, 2021.

[6] A. K. Jain, M. N. Murty und P. J. Flynn, “Data clustering: A review”, *ACM Computing Surveys*, 1999.

[7] S. P. Lloyd, “Least Squares Quantization in PCM”, *IEEE Transactions on Information Theory*, Jg. 28, Nr. 2, S. 129–137, 1982.

[8] L. Kaufman und P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990, ISBN: 978-0471878766.

[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] A. Rule, A. Birmingham, C. Zuniga, I. Altintas, S.-C. Huang, R. Knight, N. Moshiri und M. H. Nguyen, “Ten Simple Rules for Reproducible Research in Jupyter Notebooks”, *PLOS Computational Biology*, Jg. 15, Nr. 7, e1007007, 2019.

[11] J. Bois und A. Rokem, “Kaggle as a Platform for Machine Learning Education”, *Journal of Data Science Education*, Jg. 1, Nr. 1, 2021.

[12] W. Reade und A. Chow, *Tabular Playground Series – Juli 2022*, Kaggle Competition, https://kaggle.com/competitions/tabular-playground-series-jul-2022, 2022.

[13] JLU_LiuYun, *BigData & DataMining – 2nd Ex*, Kaggle Competition, https://kaggle.com/competitions/bigdata-and-datamining-2nd-ex, 2022.

[14] N. Shah, *DM-Assignment 1*, Kaggle Competition, https://kaggle.com/competitions/dmassign1, 2020.