Vergleich von häufigsten Clustering-Algorithmen auf Kaggle

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

***Zusammenfassung*—Clustering-Algorithmen gehören zu den grundlegenden Methoden der Data Science und ermöglichen es, Datenpunkte anhand von ähnlichkeiten zu gruppieren. Diese Arbeit untersucht systematisch die am häufigsten verwendeten Clustering-Algorithmen im Rahmen von Kaggle-Wettbewerben. Der Fokus liegt auf ihrer Häufigkeit, den Implementierungsdetails und den Kriterien, die die Auswahl eines Algorithmus bestimmen. Analysiert werden KMeans, DBSCAN, Gaussian Mixture, Agglomerative Clustering und Spectral Clustering.**

Der Rest muss noch ergänzt werden!

I. Einführung

CLUSTERING-Algorithmen sind eine der zentralen Techniken in der Data Science und ermöglichen die Gruppierung von Datenpunkten basierend auf Ähnlichkeiten innerhalb der Daten. Diese Methode wird in einer Vielzahl von Anwendungen wie Mustererkennung, Datensegmentierung und der Entwicklung von Vorhersagemodellen eingesetzt. Die Auswahl des geeigneten Clustering-Algorithmus hängt dabei von mehreren Faktoren ab, darunter die Datenstruktur, der spezifische Anwendungsfall und die Skalierbarkeit des Algorithmus.

Auf Plattformen wie Kaggle, einer beliebten Umgebung für Data-Science-Wettbewerbe, wird eine Vielzahl von Clustering-Algorithmen eingesetzt, um reale Analyseprobleme zu lösen. Diese Wettbewerbe bieten eine wertvolle Gelegenheit, die aktuellen Präferenzen und Trends in der Anwendung von Clustering-Methoden zu untersuchen. Gleichzeitig stellt sich die Frage, ob die auf Kaggle bevorzugten Algorithmen auch in industriellen Kontexten eine ähnlich hohe Relevanz besitzen.

Ziel dieser Arbeit ist es, die am häufigsten verwendeten Clustering-Algorithmen in den letzten 20 Kaggle-Wettbewerben systematisch zu analysieren und zu dokumentieren. Dabei werden sowohl die spezifischen Anwendungsbereiche als auch die Unterschiede zwischen den Algorithmen untersucht. Ein besonderer Fokus liegt auf der Identifikation der Faktoren, die die Wahl eines Algorithmus beeinflussen, und der Evaluation, welche dieser Ansätze auch außerhalb von Kaggle bevorzugt werden.

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.

Durch diese Analyse wird ein umfassender Überblick über die aktuellen Trends bei der Anwendung von Clustering-Algorithmen gewonnen, was sowohl für die Forschung als auch für industrielle Anwendungen von Bedeutung ist.

II. Hintergrund

A. Kaggle

Kaggle ist eine Plattform, die Wettbewerbe und den Wissensaustausch im Bereich des maschinellen Lernens ermöglicht. Sie wird von einer Vielzahl von Unternehmen von kleinen Start-ups bis hin zu großen internationalen Konzernen genutzt, um komplexe Probleme zu lösen und innovative Ansätze zu fördern. Die Plattform bietet nicht nur eine Umgebung für Data-Science-Wettbewerbe, sondern auch eine breite Palette an Ressourcen wie Datensätze, Tutorials und Tools, die sowohl Einsteigern als auch Experten zugutekommen. [11]

B. Clustering

Clustering ist eine Methode des unüberwachten maschinellen Lernens, die darauf abzielt, Datenpunkte basierend auf ihren inhärenten Ähnlichkeiten in Gruppen, sogenannte Cluster, zu unterteilen. Dieses Verfahren ermöglicht es, in Datensätzen verborgene Strukturen zu identifizieren, ohne dass vorherige Labels oder Kategorisierungen erforderlich sind. Ein exemplarisches Clustering-Ergebnis ist in Abbildung 1 dargestellt. [1]

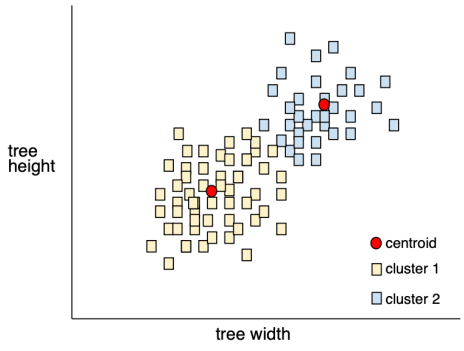

Abbildung 1. Beispiel für ein Clustering-Ergebnis, basierend auf der Darstellung von Google Developers [1]

C. Liste Clustering Algos

Die genannten Algorithmen sind etablierte Verfahren im Bereich des maschinellen Lernens und der Datenanalyse. Im Folgenden werden sie detailliert beschrieben:

- **KMeans**: Ein unüberwachter Clustering-Algorithmus, der Datenpunkte in eine vorgegebene Anzahl von Clustern unterteilt, sodass jeder Punkt dem Cluster mit dem nächstgelegenen Mittelwert (Zentroid) zugewiesen wird. Der Algorithmus minimiert die Summe der quadratischen Abstände zwischen den Punkten und ihren jeweiligen Zentroiden und eignet sich besonders für Daten mit konvexen Clusterstrukturen. [5]
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: Ein dichtebasierter Clustering-Algorithmus, der Cluster als Bereiche höherer Dichte definiert und in der Lage ist, Cluster beliebiger Form zu identifizieren sowie Ausreißer effektiv zu erkennen. DBSCAN erfordert keine vorherige Angabe der Anzahl der Cluster und kann auch in Daten mit Rauschen robuste Cluster finden. [7]
- **GaussianMixture**: Ein probabilistisches Modell, das die Daten als Mischung mehrerer normalverteilter Komponenten modelliert, wobei jeder Cluster durch eine Gaußsche Komponente repräsentiert wird. Dies ermöglicht die Modellierung von Clustern mit unterschiedlichen Formen und Größen sowie die Berechnung der Wahrscheinlichkeit, mit der ein Datenpunkt zu einem bestimmten Cluster gehört. [8]
- **AgglomerativeClustering**: Ein hierarchischer Clustering-Ansatz, der mit jedem Datenpunkt als eigenständigem Cluster beginnt und iterativ die nächstgelegenen Paare von Clustern zusammenführt, bis alle Punkte in einem einzigen Cluster vereint sind oder ein Abbruchkriterium erreicht wird. Dies erzeugt eine hierarchische Struktur von Clustern, die in einem Dendrogramm visualisiert werden kann, und ermöglicht die Analyse der Daten auf verschiedenen Granularitätsebenen. [5]
- **SpectralClustering**: Ein graphbasierter Clustering-Algorithmus, der die Datenpunkte als Knoten in einem Graphen darstellt. Die Kanten werden durch ähnlichkeitsmaße gewichtet, um die Beziehung zwischen den Datenpunkten darzustellen. Anschließend wird eine spektrale Analyse der Laplacian-Matrix des Graphen durchgeführt, um Cluster basierend auf den Eigenvektoren des Graphen zu identifizieren. Dieser Ansatz ist besonders effektiv für nichtkonvexe Clusterstrukturen und Daten mit komplexen Beziehungen. [6]

Die Wahl des geeigneten Algorithmus hängt von den spezifischen Eigenschaften des Datensatzes und den Zielen der Analyse ab. Faktoren wie die Form und Dichte der Cluster, das Vorhandensein von Rauschen und Ausreißern sowie die Skalierbarkeit des Algorithmus spielen hierbei eine entscheidende Rolle.

D. Silhouette-Bewertung

Die Silhouette-Bewertung ist ein Maß zur Beurteilung der Qualität von Clusteranalysen. Sie kombiniert die interne Kohäsion innerhalb eines Clusters mit der Trennung zu anderen Clustern. Für jeden Datenpunkt i wird der Silhouettenwert $s(i)$ wie folgt berechnet:

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

Dabei gilt:

- $a(i)$: Durchschnittliche Distanz von Punkt i zu allen anderen Punkten im selben Cluster.
- $b(i)$: Kleinste durchschnittliche Distanz von Punkt i zu allen Punkten in einem anderen Cluster, d.h., die Distanz zum nächstgelegenen Cluster.

Die Silhouettenwerte liegen im Bereich von -1 bis 1:

- Werte nahe 1 deuten darauf hin, dass der Punkt gut in seinem Cluster platziert ist und weit von anderen Clustern entfernt liegt.
- Werte um 0 weisen darauf hin, dass der Punkt an der Grenze zwischen zwei Clustern liegt.
- Negative Werte deuten darauf hin, dass der Punkt möglicherweise dem falschen Cluster zugeordnet wurde.

Der durchschnittliche Silhouettenwert aller Datenpunkte wird häufig verwendet, um die Gesamtqualität eines Clusterings zu bewerten. Ein höherer durchschnittlicher Silhouettenwert zeigt eine bessere Clusterstruktur an. Dieses Konzept wurde von Peter J. Rousseeuw in seinem 1987 veröffentlichten Artikel eingeführt. [10]

III. Mehtodik

In diesem Abschnitt wird die geplante Methodik zur Analyse von Wettbewerben auf Kaggle im Hinblick auf Clustering-Algorithmen beschrieben. Die Methodik umfasst die folgenden Schritte:

1) **Datenbeschaffung:** Extrahieren der letzten 20 Wettbewerbe auf Kaggle, wobei der Fokus auf solchen Wettbewerben liegt, die Clustering-Algorithmen beinhalten.
2) **Notebook-Analyse:** Analyse der zugehörigen Kaggle-Notebooks durch die Suche nach relevanten Schlagwörtern im Zusammenhang mit Clustering-Algorithmen. Dies umfasst Algorithmenamen (z. B. k-means, DBSCAN, hierarchisches Clustering) sowie verwandte Begriffe (z. B. Clustering-Bewertungsmetriken).
3) **Vergleich der Algorithmen:** Vergleich der identifizierten Clustering-Algorithmen im Hinblick auf ihre Unterschiede, Häufigkeit der Verwendung und Implementierungsdetails.

Zur Effizienzsteigerung und Automatisierung wird ein Python-Skript eingesetzt, das folgende Aufgaben übernimmt:

- Automatischer Download der Notebooks aus den identifizierten Kaggle-Wettbewerben.
- Parsen und Durchsuchen der Inhalte nach Clustering-relevanten Schlüsselwörtern.

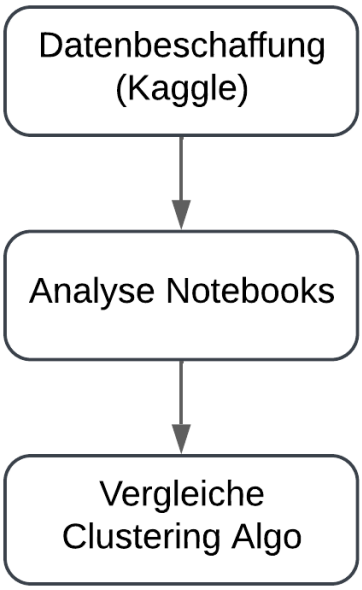

Abbildung 2. Ablaufdiagramm, eigene Darstellung

- Aggregation und Organisation der Ergebnisse für die weitere Analyse.

Dieses Vorgehen bietet einen systematischen Rahmen, um Trends zu identifizieren und den Einsatz von Clustering-Algorithmen in realen Datensätzen und Wettbewerben zu bewerten.

Den Source Code dazu ist unter folgendem Link zu finden: https://github.com/XplorodoX/BigData

IV. Analyse der eingesetzten Clustering-Algorithmen

Die Folgende Abbildung 3 zeigt die Verteilung der Clustering Algos auf die einzelnen Notebooks verteilt. Wie man dabei sieht ist hier K-Means bei allen Wettbewerben vertreten und verwendet. Dabei fällt auch auf das von 20 Wettbewerben nur 14 Wettbewerbe sind die in der Grafik gelistet werden, da

Abbildung 4 zeigt die Häufigkeit der Algorithmen innerhalb einzelner Wettbewerbe. Es zeigt sich, dass bestimmte Algorithmen, wie beispielsweise der *RandomForestClassifier*, deutlich häufiger verwendet werden als andere.

Der K-Means-Algorithmus war in der Analyse der am häufigsten verwendete Algorithmus und wurde in 25 der untersuchten Notebooks eingesetzt. Darauf folgte eine weitere Variante des K-Means, die in mehr als fünf Notebooks vertreten war. Der DBSCAN-Algorithmus belegte den dritten Platz, gefolgt von Agglomerative Clustering. Der Gaussian Mixture Clustering-Algorithmus wurde am seltensten verwendet.

Aus der Analyse ergibt sich, dass die fünf am häufigsten verwendeten Algorithmen die folgenden sind:

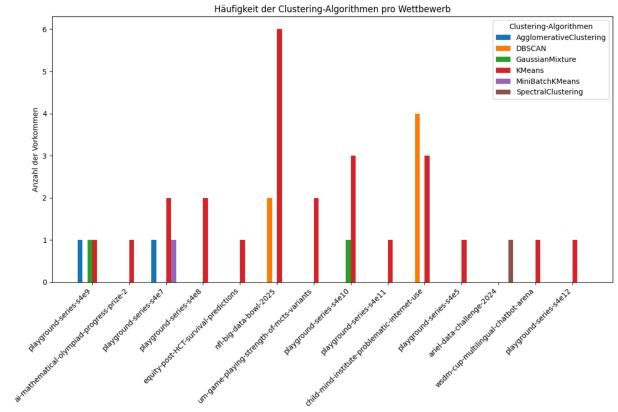

Abbildung 3. Häufigkeit der Clustering-Algorithmen pro Wettbewerb, eigene Darstellung.

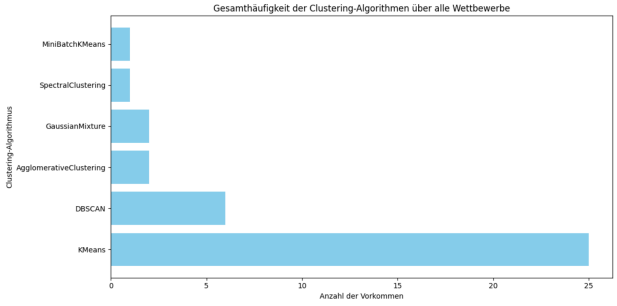

Abbildung 4. Häufigkeit der Clustering-Algorithmen pro Wettbewerb, eigene Darstellung.

1) K-Means
2) DBSCAN
3) AgglomerativeClustering
4) GaussianMixture
5) SpectralClustering

Diese Algorithmen werden im folgenden Kapitel miteinander verglichen, um ihre Vor- und Nachteile sowie Einsatzmöglichkeiten näher zu beleuchten.

V. Vergleich der Clustering-Algorithmen

Für den Vergleich der Clustering-Algorithmen wurden Beispielwerte genutzt, um die Unterschiede zwischen den einzelnen Methoden anschaulich darzustellen. Die Umsetzung erfolgte mit der `scikit-learn`-Bibliothek in Python, die verschiedene Clustering-Algorithmen bereitstellt. Ziel ist es, die Besonderheiten der Algorithmen zu erklären und ihre Ergebnisse anhand der Qualität der gebildeten Gruppen (Cluster) zu bewerten.

A. KMeans

Abbildung 5 zeigt die Ergebnisse des *KMeans*-Algorithmus, der eine Silhouette-Bewertung von 0.48 erreicht hat. KMeans zeichnet sich durch eine einfache Implementierung und hohe Effizienz in der Ausführung aus. Der Algorithmus erzeugt klare und gut definierte kugelförmige Cluster. Dies wird in den Clustern der Abbildung deutlich, die eine klare Trennung aufweisen. Allerdings zeigt sich, dass KMeans bei nichtlinearen

Datenstrukturen an seine Grenzen stößt, da der Algorithmus von kugelförmigen Clustergeometrien ausgeht.

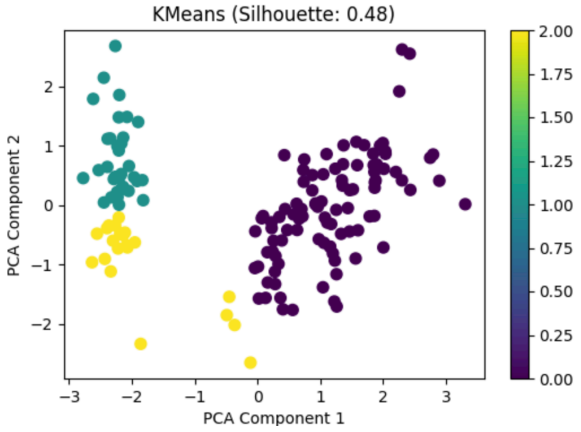

Abbildung 5. Anwendung von KMeans-Clustering-Algorithmen in Wettbewerben, eigene Darstellung.

B. DBSCAN

Die Ergebnisse des *DBSCAN*-Algorithmus sind in Abbildung 6 dargestellt. Mit einer Silhouette-Bewertung von 0.36 weist DBSCAN die niedrigste Clustertrennung unter den getesteten Algorithmen auf. Seine Stärke liegt in der Identifikation von dichten Regionen sowie der Fähigkeit, Ausreißer zu ignorieren. Die Abbildung verdeutlicht jedoch, dass die Wahl der Parameter großen Einfluss auf das Ergebnis hat, was die Anwendbarkeit erschweren kann.

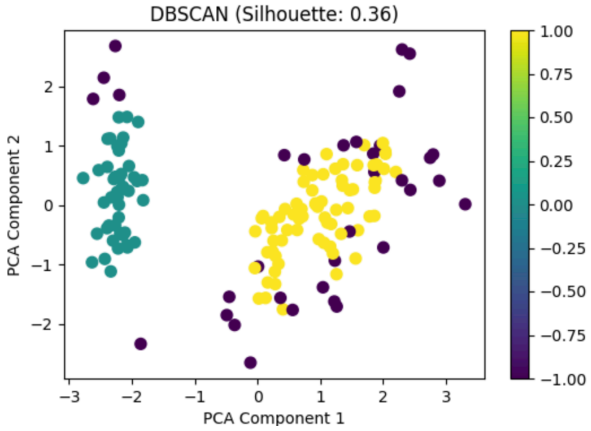

Abbildung 6. Verwendung von DBSCAN-Clustering-Algorithmen in verschiedenen Wettbewerben, eigene Darstellung.

C. Gaussian Mixture Model

Abbildung 7 illustriert die Ergebnisse des *Gaussian Mixture Model* (GMM). Dieser Algorithmus erzielte, ähnlich wie KMeans, eine Silhouette-Bewertung von 0.48. GMM modelliert die Cluster als multivariate Normalverteilungen und kann somit auch überlappende Cluster gut abbilden. Die Abbildung zeigt, dass die Cluster klar getrennt sind, wobei eine gewisse Flexibilität in der Form der Cluster erkennbar ist. Die Voraussetzung normalverteilter Daten kann jedoch in realen Szenarien eine Einschränkung darstellen.

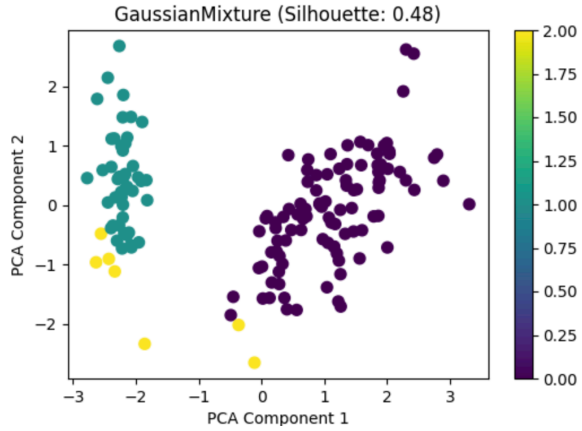

Abbildung 7. Häufigkeit der Anwendung von Gaussian Mixture Modelling (GMM) in Wettbewerben, eigene Darstellung.

D. Spectral Clustering

Die Resultate des *Spectral Clustering* sind in Abbildung 8 dargestellt. Mit einer Silhouette-Bewertung von 0.46 liegt der Algorithmus im Mittelfeld. Spectral Clustering ist besonders geeignet für nichtlineare Datenstrukturen und liefert hier klar definierte Cluster. Die Abbildung verdeutlicht jedoch, dass die Trennung zwischen den Clustern nicht so scharf ist wie bei KMeans oder GMM. Zudem ist der Algorithmus recheninten-siv und weniger skalierbar.

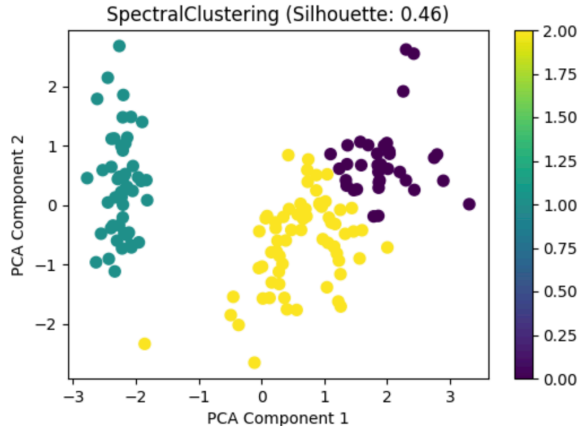

Abbildung 8. Einsatz von Spectral Clustering-Algorithmen in Wettbewerben, eigene Darstellung.

E. Agglomerative Clustering

In Abbildung 9 werden die Ergebnisse des *Agglomerative Clustering* gezeigt, das eine Silhouette-Bewertung von 0.45 erreicht. Dieser hierarchische Algorithmus verbindet iterativ Datenpunkte zu Clustern. Wie die Abbildung zeigt, sind die

Cluster relativ gut getrennt, allerdings ähnelt die Clusterstruktur stark der von Spectral Clustering. Der Algorithmus eignet sich gut für hierarchische Analysen, ist jedoch anfällig für Ausreißer.

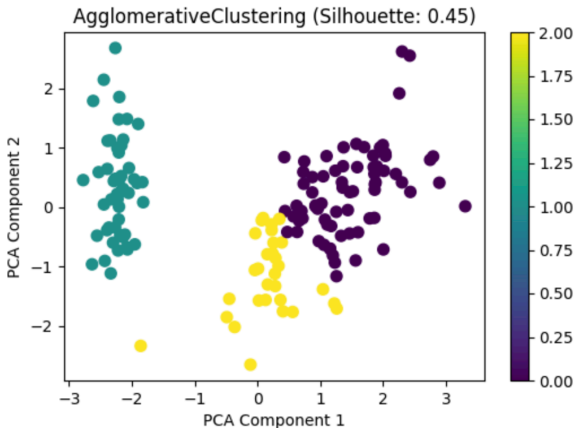

Abbildung 9. Verwendung von Agglomerative Clustering-Algorithmen in verschiedenen Wettbewerben, eigene Darstellung.

Abbildung 4 zeigt, dass *KMeans* der am häufigsten eingesetzte Algorithmus in Wettbewerben ist, was auf seine Einfachheit und Effizienz zurückzuführen ist. *DBSCAN* hingegen wird seltener verwendet, obwohl er in spezifischen Anwendungsfällen, wie der Erkennung von Ausreißern, wertvolle Ergebnisse liefern kann. *Gaussian Mixture Models* und *Spectral Clustering* bieten flexible Ansätze, sind jedoch rechenintensiver. *Agglomerative Clustering* stellt eine robuste Wahl für hierarchische Daten dar, ist aber weniger flexibel bei komplexen Datenstrukturen.

VI. Fazit

...

Literatur

[1] Google Developers, *Clustering-Algorithmen — Machine Learning*, verfügbar unter: https://developers.google.com/machine-learning/clustering/clustering-algorithms?hl=de, Zugriff am 9. Dezember 2024.

[2] Milecia McGregor, *8 Clustering Algorithms in Machine Learning that All Data Scientists Should Know*, FreeCodeCamp, 21. September 2020, verfügbar unter: https://www.freecodecamp.org/news/8-clustering-algorithms-in-machine-learning-that-all-data-scientists-should-know/, Zugriff am 9. Dezember 2024.

[3] GeeksforGeeks, *Clustering in Machine Learning*, verfügbar unter: https://www.geeksforgeeks.org/clustering-in-machine-learning/, Zugriff am 9. Dezember 2024.

[4] Dingding Xu und Yingjie Tian, *A Comprehensive Survey of Clustering Algorithms*, Annals of Data Science, Band 2, Nummer 2, 2015, Seiten 165193.

[5] Scikit-learn developers, *scikit-learn: Machine Learning in Python*, verfügbar unter: https://scikit-learn.org/stable/, Zugriff am 9. Dezember 2024.

[6] Scikit-learn developers, *SpectralClustering scikit-learn 1.5.2 documentation*, verfügbar unter: https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.SpectralClustering.html, Zugriff am 9. Dezember 2024.

[7] Andrew Tate, *Comparing Density-Based Methods*, Hex, 24. Oktober 2023, verfügbar unter: https://hex.tech/blog/comparing-density-based-methods/, Zugriff am 9. Dezember 2024.

[8] Nate Rosidi, *Clustering with scikit-learn: A Tutorial on Unsupervised Learning*, KDnuggets, 11. Mai 2023, verfügbar unter: https://www.kdnuggets.com/2023/05/clustering-scikitlearn-tutorial-unsupervised-learning.html, Zugriff am 9. Dezember 2024.

[9] Scikit-learn developers, *Selecting the number of clusters with silhouette analysis on KMeans clustering*, verfügbar unter: https://scikit-learn.org/1.5/auto_examples/cluster/plot_kmeans_silhouette_analysis.html, Zugriff am 9. Dezember 2024.

[10] Rousseeuw, Peter J., *Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis*, Journal of Computational and Applied Mathematics, Vol. 20, pp. 53–65, Elsevier, 1987.

[11] BigData-Insider, *Was ist Kaggle?*, verfügbar unter: https://www.bigdata-insider.de/was-ist-kaggle-a-951812/, Zugriff am 9. Dezember 2024.