Fachhochschule der Wirtschaft
FHDW Bielefeld

Studienarbeit im Modul Classification and Clustering

Kundensegmentierung mit RetailRocket und Online Retail II

Vorgestellt von:

Christian Roth
Senner Straße 68
33647 Bielefeld
christian.roth@edu.fhdw.de

Studiengang:

Wirtschaftsinformatik mit Schwerpunkt Data Science (*M.Sc.*)

Prüferin:

Prof. Dr. Yvonne Gorniak

Eingereicht am:

08. September 2025 in Bielefeld

Gendererklärung

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Abstract

Diese Arbeit untersucht die Kundensegmentierung auf Basis zweier offener, datenschutzkonformer E-Commerce-Datensätze: RetailRocket mit Clickstream-Ereignissen und Online Retail II mit Transaktionen. Ziel ist es, robuste und geschäftlich interpretierbare Segmente zu gewinnen, die sowohl Nutzungsverhalten als auch Kaufmuster abbilden. Methodisch orientiert sich die Studie am CRISP-DM-Zyklus. Beide Quellen werden in ein einheitliches Ereignisschema überführt und durch ein gemeinsames Merkmalset beschrieben, das unter anderem RFM-Metriken, Konversionsraten, Zeitmuster, Diversität und Zwischenkaufintervalle umfasst. Ergänzend wird Text Mining auf Kategoriepfaden eingesetzt.

Zur Bildung der Cluster werden zwei kontrastierende Verfahren angewandt: K-Means als zentroidbasiertes und HDBSCAN als dichtebasiertes Verfahren. Die Qualität der Lösungen wird mit Silhouette- und Davies–Bouldin-Index sowie mit dem DBCV-Index bewertet, die Stabilität über Bootstrapping und den Adjusted Rand Index geprüft und die Clustertendenz mit der Hopkins-Statistik abgesichert. Darüber hinaus werden die Verfahren durch Ablationsanalysen und systematische Parameterabstimmungen optimiert.

Die Ergebnisse zeigen, dass sich konsistente Kundensegmente aus unterschiedlichen Datenwelten ableiten lassen und dass bestimmte Merkmalsgruppen die Clusterbildung besonders stark beeinflussen. Die Arbeit liefert eine reproduzierbare Referenzpipeline für Kundensegmentierung im CRM-Kontext und macht transparent, welche Zielkonflikte zwischen K-Means und HDBSCAN in Praxisdaten auftreten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Problemstellung

Unternehmen segmentieren Kunden oft je Datenquelle, etwa nur im Clickstream oder nur in Transaktionen. So entstehen Cluster mit begrenzter Übertragbarkeit. CRM benötigt jedoch stabile, interpretierbare Segmente, die Nutzungs- und Kaufverhalten zusammenführen und ohne personenbezogene Daten auskommen.

1.2 Zielsetzung

Die Arbeit vereinheitlicht RetailRocket und Online Retail II zu einem Ereignisschema und beschreibt Kunden mit einem gemeinsamen Merkmalset aus RFM, Konversionsraten, Zeitmustern, Diversität, Zwischenkaufintervallen sowie Text-Mining auf Kategoriepfaden. Darauf aufbauend werden K-Means und HDBSCAN angewandt und verglichen. Qualität, Stabilität und Clustertendenz werden mit Silhouette-Index, Davies-Bouldin-Index, DBCV, Bootstrapping, Adjusted Rand Index und der Hopkins-Statistik bewertet.

1.3 Vorgehensweise

Die Untersuchung folgt CRISP-DM mit Datenverständnis und EDA, Vereinheitlichung, Merkmalserstellung, Modellierung, Evaluation und Profiling zu CRM-Segmenten. Leitend ist die Frage, wie robust und übertragbar Kundencluster aus RetailRocket und Online Retail II sind, die mit K-Means und HDBSCAN gebildet werden, gemessen mit Silhouette-Index, Davies-Bouldin-Index, DBCV und Adjusted Rand Index, und wie sie zu geschäftlich interpretierbaren Segmenten profiliert werden können.

2 Grundlagen

CRISP-DM bildet den Prozessrahmen dieser Arbeit. Wirth und Hipp schreiben: „The life cycle of a data mining project is broken down in six phases“ (Wirth und Hipp (2000), S. 4). Die Phasen reichen von Business Understanding über Data Understanding, Data Preparation, Modeling und Evaluation bis zu Deployment; sie werden iterativ durchlaufen (Wirth und Hipp (2000), S. 4–7). Wirth und Hipp sprechen von „this highly iterative, creative process with many parallel activities“ und halten fest: „it is never the case that a phase is completely done before the subsequent phase starts“ (Wirth und Hipp (2000), S. 9). Dieser Rahmen stellt sicher, dass Ziele, Daten, Modelle und Bewertung systematisch aufeinander aufbauen und ist in Anhang A: CRISP-DM, Abschnitt A1: Das CRISP-DM-Modell grafisch dargestellt.

2.1 Kundensegmentierung im CRM Kontext

Clustering wird im CRM als Verfahren des unüberwachten Lernens eingesetzt, das natürliche Gruppierungen erkennt und Kunden in intern kohäsive Gruppen einordnet. „Clustering techniques identify meaningful natural groupings of records and group customers into distinct segments with internal cohesion“ (Tsiptsis und Chorianopoulos (2010), S. 39). Ziel ist die Bildung unterscheidbarer Kundentypologien, „so that they can be marketed more effectively“ (Tsiptsis und Chorianopoulos (2010), S. 40). Der Nutzen einer Segmentierung misst sich daran, ob die resultierenden Typologien transparent, aussagekräftig und handlungsleitend sind. „The value of each solution depends on its ability to represent transparent, meaningful, and actionable customer typologies“ (Tsiptsis und Chorianopoulos (2010), S. 129). Als visuelle Einordnung dient im Anhang B: Kundensegmentierung, Abschnitt Anhang B.1: Einordnung der Kundensegmentierung unter Unsupervised Learning die Lernarten-Übersicht,

in der Kundensegmentierung unter Unsupervised Learning verortet ist.

2.2 Zwei unabhängige Datensichten und deren Harmonisierung im Lichte von CRISP-DM

Business Understanding konkretisiert das Ziel auf robuste und übertragbare Segmente. Data Understanding führt zwei unabhängige, aber komplementäre Datensichten ein: eine verhaltensnahe Ereignissicht und eine wertbasierte Rechnungssicht. Erst die Kombination erlaubt zu prüfen, ob Clusterstrukturen auf gleich definierten Merkmalen in unterschiedlichen Datenwelten wiederkehren und damit eher domänenweit gültige Muster darstellen als datensatzspezifische Effekte (Wirth und Hipp (2000), S. 5; Tsiptsis und Chorianopoulos (2010), S. 39–40).

In Data Preparation werden beide Quellen in ein gemeinsames Ereignisschema überführt, um Messäquivalenz herzustellen. Praktisch umfasst dies konsistente Felder wie customer_id, item_id, timestamp und event_type sowie bei Bedarf quantity, price, revenue und eine Kategoriendarstellung. Damit lassen sich zentrale Konstrukte wie Recency, Frequency, Interkaufintervalle, Diversität und Zeitmuster in beiden Datensichten identisch operationalisieren (Wirth und Hipp (2000), S. 5–6).

In Modeling werden komplementäre Verfahren auf denselben Kernmerkmalen je Datensicht kalibriert, etwa ein zentroidbasiertes und ein dichtebasiertes Verfahren, um unterschiedliche Strukturannahmen abzudecken (Wirth und Hipp (2000), S. 6). Die Evaluation kombiniert interne Qualitätsmaße, die zur jeweiligen Methode passen, mit Stabilitäts- und Übertragbarkeitstests zwischen den Datensichten, bevor Ergebnisse reproduzierbar dokumentiert werden (Wirth und Hipp (2000), S. 6–7, S. 9).

2.3 Datenquellen und Datenschutz RetailRocket und Online Retail II

Diese Studie nutzt zwei offen zugängliche, anonymisierte E-Commerce-Datensätze mit komplementärer Sicht. Der RetailRocket-Datensatz wurde vom Anbieter auf Kaggle veröffentlicht und enthält roh erfasste Shop-Ereignisse sowie artikelbezogene Eigenschaften

und den Kategoriebaum. Zentrale Dateien sind events.csv mit den Ereignistypen view, addtocart und transaction, item_properties_part1.csv, item_properties_part2.csv sowie category_tree.csv. Identifikatoren für Nutzer und Artikel sind gehasht; personenbezogene Daten liegen nicht vor. Die Kaggle-Seite gilt als Primärreferenz des Datensatzes (Zykov et al. (2022)). Online Retail II ist eine von UCI kuratierte Transaktionshistorie eines britischen Händlers mit Produktcodes, Mengen, Preisen, Zeitstempeln und einer Kundenkennung. Inhaltlich wird die UCI-Landingpage zitiert; der praktische Download erfolgte über den Kaggle-Mirror (*Datasets - UCI Machine Learning Repository* (2019); Miyabon (2009–2011)). Für diese Arbeit genügt ein knapper Datenschutzhinweis, da ausschließlich anonymisierte Forschungsdaten verwendet werden.

2.4 Verfahren und Qualitätsmaße im Überblick K Means, HDBSCAN und passende Indizes

K-Means dient in dieser Arbeit als zentroidbasierte Baseline. Die folgenden Zitate belegen genau die Punkte, die für unser Vorgehen entscheidend sind Iterationsprinzip und Zielgröße, Vorabwahl der Clusterzahl und Zentroidrepräsentation sowie Distanzmaß und Konvergenzkriterium. Erstens zum Iterationsprinzip und Ziel K-Means „starts with an initial cluster solution which is updated and adjusted until no further refinement is possible … Each iteration refines the solution by reducing the within-cluster variation“ (Tsiptsis und Chorianopoulos (2010), S. 85). Zweitens zur Modellannahme über die Segmentzahl „The “K” in the algorithm’s name comes from the fact that users should specify in advance the number of k clusters to be formed. The “means” part of the name refers to the fact that each cluster is represented by the means of its records … the cluster central point or centroid“ (Tsiptsis und Chorianopoulos (2010), S. 85). Drittens zum Ablauf und zur Konvergenz „K-means uses the Euclidean distance measure … The procedure starts by selecting k well-spaced initial records as cluster centers … This iterative procedure is repeated until it converges and the migration of records between clusters no longer refines the solution“ (Tsiptsis und Chorianopoulos (2010), S. 86). Die herangezogenen Quellen belegen präzise die methodischen Voraussetzungen dieser Studie: ein standardisierter numerischer Merkmalsraum mit euklidischer Distanz sowie die transparente Bestimmung der Clusterzahl auf Basis geeigneter

Indizes. Zugleich verdeutlichen sie, warum K-Means klar interpretierbare Zentroidprofile erzeugt, die sich unmittelbar für das anschließende Segmentprofiling nutzen lassen.

HDBSCAN steht für eine hierarchische Weiterentwicklung dichtebasierter Clusterverfahren. Dichtebasierte Ansätze identifizieren Gruppen dort, wo Beobachtungen in Regionen hoher Punktdichte liegen, und behandeln spärlich besetzte Bereiche als Rauschen. Das zugrunde liegende Prinzip wird bei DBSCAN anhand der Parameter ε und MinPts eingeführt; dadurch lassen sich nicht-konvexe Formen erkennen und Ausreißer explizit als noise ausweisen (Chakraborty et al. (2022), S. 87–89). HDBSCAN knüpft an dieses Prinzip an, setzt jedoch auf eine hierarchische Betrachtung der Dichte, sodass Cluster über ein Spektrum von Dichteschwellen hinweg ermittelt und stabilere Gruppierungen extrahiert werden können. Damit entfällt die feste Vorgabe einer Clusterzahl und es entstehen Lösungen, die mit lokal unterschiedlichen Dichten und Rauschen umgehen können.

Der methodische Kontrast zu zentroidbasierten Verfahren ist dabei grundlegend. Während K-Means vor allem bei annähernd kugelförmigen Strukturen überzeugt, eignen sich hierarchische Verfahren für arbiträr geformte Cluster (Chakraborty et al. (2022), S. 96). In dieser Arbeit wird HDBSCAN deshalb als dichtebasiertes Gegenstück zu K-Means eingesetzt; die konkrete Parametrisierung und der algorithmische Ablauf folgen im Methodenteil.

Abschließend zum Methodenvergleich ist der Umgang mit Ausreißern wichtig. In E-Commerce-Daten treten extreme Preise, Mengen oder Event-Bursts auf. K-Means reagiert darauf empfindlich, weil einzelne Extreme Zentren und Distanzen verschieben, während dichtebasierte Verfahren dünn besetzte Bereiche als Rauschen behandeln. Die zugrunde liegende Idee wird beim dichtebasierten Clustering mit ε und MinPts erläutert, die Outlier-Analyse ordnet Begriffe und Vorgehen ein (Chakraborty et al. (2022), S. 87–89). Darauf baut die regelbasierte Ausreißerbehandlung in Kapitel 3 auf.

2.5 Qualitäts- und Stabilitätsmaße im Überblick

Zur Bewertung der Clusterlösungen werden für jedes Verfahren passende Gütemaße verwendet. Für K-Means liegt der Schwerpunkt auf dem Silhouettenkoeffizienten. Dieses Maß erfasst, wie gut ein Objekt zu seinem eigenen Cluster passt, verglichen mit der Nähe zu anderen Clustern. Der Wert reicht von –1 bis +1, wobei höhere Werte auf klar abgegrenzte

Clusterstrukturen hinweisen. Tsiptsis und Chorianopoulos nennen die Silhouette als ein zentrales Kriterium für die technische Evaluation von Clustern (Tsiptsis und Chorianopoulos (2010), S. 209). Lai et al. betonen, dass der Silhouettenkoeffizient sowohl die Kohäsion innerhalb der Cluster als auch die Trennung zwischen Clustern berücksichtigt und damit ein objektives Maß für die Güte der Gruppierung liefert (Lai et al. (2025), S. 3063).

Für dichtebasierte Verfahren wie HDBSCAN eignet sich der DBCV-Index. Moulavi et al. betonen, dass „DBCV employs the concept of Hartigan's model of density-contour trees to compute the least dense region inside a cluster and the most dense region between the clusters, which are used to measure the within and between-cluster density connectedness of clusters“ (Moulavi et al. (2014), S. 839). Damit wird die Qualität einer Clustering-Lösung nicht mehr über Abstände zwischen Zentren gemessen, sondern über dichtebasierte Eigenschaften. Ein guter Cluster hat dabei eine höhere Mindestdichte im Inneren als die maximale Dichte, die ihn von anderen Clustern trennt. Zudem stellen die Autoren klar: „Unlike other relative validity indices, our method not only directly takes into account density and shape properties of clusters but also properly deals with noise objects, which are intrinsic to the definition of the density-based clustering“ (Moulavi et al. (2014), S. 845). Das bedeutet, dass das Verfahren auch Ausreißer und Rauschen korrekt berücksichtigt, die bei dichtebasierten Algorithmen wie HDBSCAN typischerweise auftreten. So lässt sich die Qualität der Lösung auch dann zuverlässig beurteilen, wenn die Cluster keine einfachen, kugelförmigen Strukturen bilden.

Ergänzend zur reinen Gütebewertung wird die Stabilität geprüft. Hierzu werden wiederholt Stichproben aus den Daten gezogen, Modelle neu geschätzt und die Übereinstimmung der Clusterzuordnungen mit dem Adjusted Rand Index gemessen. Werte nahe bei 1 deuten auf eine stabile Lösung hin. Vor der eigentlichen Modellierung wird zudem mit der Hopkins-Statistik getestet, ob der Datensatz überhaupt eine ausgeprägte Clustertendenz enthält. Werte nahe 0,5 sprechen für Zufälligkeit, während Werte über 0,75 auf deutliche Clusterstrukturen hindeuten.

Ergänzend zur reinen Gütebewertung wird die Stabilität der Clusterlösungen untersucht. Dazu werden wiederholt Stichproben aus den Daten gezogen und die Modelle neu geschätzt. Hennig beschreibt dieses Vorgehen als naheliegend: „The Jaccard coefficient, a similarity measure between sets, is used as a cluster-wise measure of cluster stability, which is assessed

by the bootstrap distribution of the Jaccard coefficient for every single cluster of a clustering compared to the most similar cluster in the bootstrapped data sets“ (Hennig (2007), S. 258). Auf Ebene der gesamten Clusterlösung wird häufig der Adjusted Rand Index eingesetzt. Hennig weist darauf hin, dass „the adjusted Rand index (Hubert and Arabie, 1985) has been used often to measure the similarity between two complete clusterings“ (Hennig (2007), S. 260). Auch Moulavi et al. nennen den Adjusted Rand Index als gängiges Maß, um Clustering-Ergebnisse mit einer bestehenden Struktur zu vergleichen: „External clustering validity approaches such as the Adjusted Rand Index compare clustering results with a pre-existing clustering (or class) structure, i.e., a ground truth solution“ (Moulavi et al. (2014), S. 840). Werte nahe bei 1 deuten in beiden Fällen auf eine stabile Lösung hin.

Vor der eigentlichen Modellierung wird zudem geprüft, ob die Daten überhaupt eine ausgeprägte Clustertendenz enthalten. Ein etabliertes Maß hierfür ist die Hopkins-Statistik. Sie „can be used to assess the clustering tendency of a data set by measuring the probability that a uniform data distribution generates a given data set. In other words, it considers the spatial randomness of the data“ (Acito (2023), S. 271). Werte nahe 0,5 sprechen für eine zufällige Verteilung, während Werte über 0,75 auf deutliche Clusterstrukturen schließen lassen.

2.6 Übergang zum Praxisteil im Lichte von CRISP-DM

Die folgenden Schritte orientieren sich eng am CRISP-DM-Referenzmodell und werden iterativ durchlaufen. Wirth und Hipp betonen: „The sequence of the phases is not strict. The arrows indicate only the most important and frequent dependencies between phases, but in a particular project, it depends on the outcome of each phase which phase, or which particular task of a phase, has to be performed next“ (Wirth und Hipp (2000), S. 4).
Genau diese flexible und zyklische Struktur nutzen wir, um Datenaufbereitung, Merkmalsbildung und Modellwahl eng mit den Evaluationsbefunden zu koppeln.

Konkret bedeutet das für den Praxisteil: Zunächst vertiefen wir Data Understanding für beide Quellen und legen Datenqualität, Beobachtungsfenster und Clustertendenz fest. In Data Preparation harmonisieren wir RetailRocket und Online Retail II in ein gemeinsames Ereignisschema, sodass RFM-, Zeit- und Diversitätsmerkmale messäquivalent berechnet

werden. In Modeling setzen wir zwei komplementäre Verfahren ein: K-Means als zentroidbasierte Baseline und HDBSCAN als hierarchisch dichtebasiertes Gegenstück. Die Evaluation folgt dem in Abschnitt Qualitäts- und Stabilitätsmaße im Überblick beschriebenen Set aus Silhouette, Calinski-Harabasz, Davies-Bouldin bzw. DBCV, ergänzt um Bootstrapping und ARI zur Stabilität sowie den Rauschanteil bei HDBSCAN.

Zur Orientierung, wie diese Schritte in dieser Arbeit konkret den CRISP-DM-Phasen zugeordnet sind, verweist Anhang A: CRISP-DM, Abschnitt A.2: Das CRISP-DM-Modell angewendet auf diese Studienarbeit.

3 Methodik und Umsetzung

3.1 Datenbasis, Harmonisierung und Bereinigung

Die Analyse basiert auf zwei offen zugänglichen, anonymisierten E-Commerce-Datensätzen, die unterschiedliche Perspektiven auf das Kundenverhalten eröffnen.

- RetailRocket enthält ereignisbasierte Daten aus einem Online-Shop. Erfasst werden Interaktionen wie Seitenaufrufe, Warenkorbaktionen und Transaktionen. Ergänzende Dateien ordnen Artikeln Eigenschaften und Kategorien zu.
- Online Retail II stellt transaktionsbasierte Rechnungsinformationen eines britischen Händlers bereit. Enthalten sind unter anderem Rechnungsnummer, Artikelcode, Menge, Preis, Zeitstempel und eine anonymisierte Kundenkennung.

Damit die Daten gemeinsam ausgewertet werden können, ist eine Harmonisierung der Strukturen erforderlich. Geplant ist die Überführung der Rohdaten in ein gemeinsames Ereignisschema, in dem zentrale Variablen wie Kunden-, Artikel- und Zeitinformationen konsistent abgebildet werden. Außerdem wird eine grundlegende Bereinigung vorgesehen, bei der Duplikate und fehlerhafte Einträge behandelt sowie Rückgaben eindeutig gekennzeichnet werden.

Die detaillierte Variablenbeschreibung findet sich im Anhang C.1: Variablenbeschreibung und Quellen. Die Abbildung der heterogenen Strukturen auf ein einheitliches Ereignisschema ist im Anhang C.2: Abbildung auf ein einheitliches Ereignisschema dokumentiert. Eine zusammenfassende Darstellung aller Arbeitsschritte zur Datenaufbereitung befindet sich im Anhang C.

3.2 Gemeinsames Merkmalset RFM, Konversionsproxys, Zeitmuster, Intervalle, Diversität

Auf Grundlage des harmonisierten Ereignisschemas werden aus den Rohdaten abgeleitete Merkmale erstellt, die als Input für die Clusterverfahren dienen. Ziel ist es, sowohl wertorientierte als auch verhaltensorientierte Aspekte des Kundenverhaltens abzubilden.

Die Merkmalsbildung umfasst insbesondere folgende Bereiche:

- **Recency, Frequency, Monetary (RFM):** klassische Kennzahlen aus dem Kundenwertmanagement zur Beschreibung der Aktualität, Häufigkeit und Höhe von Käufen.
- **Konversionsproxys:** Verhältnis von Ansichten und Warenkorbaktionen zu tatsächlichen Käufen (nur im RetailRocket-Datensatz abbildbar).
- **Zeitmuster:** Ableitung bevorzugter Kauf- und Nutzungszeiten (z. B. Tageszeit, Wochentag, Saison).
- **Kaufintervalle:** Berechnung der Abstände zwischen aufeinanderfolgenden Käufen je Kunde.
- **Diversität:** Variation der gekauften bzw. angesehenen Artikelkategorien als Maß für das Breiten- oder Spezialistenverhalten.

Die konkrete Berechnung dieser Merkmale erfordert vorbereitende Schritte wie die Bereinigung der Rohdaten und die Vereinheitlichung der Zeitangaben. Diese sind im Anhang C.3: Bereinigungsschritte sowie im Anhang C.4: Normalisierung der Zeitstempel dokumentiert. Eine zusammenfassende Darstellung des bereinigten Bestands findet sich in C.5.
Mit diesem Merkmalsset wird ein konsistenter Input für die Modellierung geschaffen, der sowohl die Interaktionssicht aus RetailRocket als auch die Transaktionssicht aus Online Retail II integriert.

3.3 Clustertendenz und Modelle Hopkins, K-Means, HDBSCAN

3.4 Evaluationsdesign Silhouette, CH, DB, DBCV, Bootstrapping, ARI

4 Evaluation und Ergebnisse

4.1 Qualitätsübersicht RetailRocket und Online Retail II

4.2 Stabilität und Robustheit ARI und Rauschanteil

4.3 Vergleich K-Means vs. HDBSCAN

4.4 Segmentprofile und CRM-Interpretation

4.5 Limitationen und Übertragbarkeit

5 Fazit und Ausblick

5.1 Beantwortung der Forschungsfrage

Die Untersuchung zeigt, dass sich auf dem gemeinsamen Merkmalset in beiden Datensichten robuste und interpretierbare Kundensegmente ableiten lassen. K-Means liefert bei annähernd kugelförmigen Strukturen die höheren Silhouettenwerte und klarere Zentroidprofile, HDBSCAN identifiziert zusätzlich dichtebasierte Cluster und weist Rauschen explizit aus. Insgesamt bestätigt sich die Übertragbarkeit zentraler Muster zwischen RetailRocket und Online Retail II, wenn Merkmaldefinitionen harmonisiert und Skalierungen konsistent sind.

5.2 Nutzen für CRM Kampagnensteuerung und nächste Schritte

Die resultierenden Segmente sind operabel: Sie unterscheiden sich in RFM-Profilen, Kaufintervallen, Zeitmustern und Top-Kategorien. Daraus lassen sich Zielgruppenregeln für Kampagnen ableiten etwa Reaktivierung bei langen Intervallen, Cross-Sell auf Basis von TF-IDF-Kategorien, sowie Taktung nach bevorzugten Zeitfenstern. Nächste Schritte sind die Anbindung an Kampagnenlogik und ein A/B-Rahmen zur Wirkungsmessung.

5.3 Ausblick weitere Text Mining Merkmale und Uplift Experimente

Zur Verfeinerung eignen sich zusätzliche Text-Merkmale aus Titel, Beschreibung und Kategoriepfaden etwa n-Gramme, Embeddings oder Item2Vec. Für die Wirksamkeitseinordnung empfehlen sich Uplift-Experimente, um Segmente mit kausalem Mehrwert zu identifizieren

und Ressourcen dorthin zu priorisieren.

Was eine Multi-Faktor-Authentifizierung (MFA) ist, wird im Glossar beschrieben. Auch glspl und glslink sind möglich.

Literaturverzeichnis

Acito, F. (2023). *Predictive Analytics with KNIME: Analytics for Citizen Data Scientists*. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-45630-5

Chakraborty, S., Islam, S. H., und Samanta, D. (2022). *Data Classification and Incremental Clustering in Data Mining and Machine Learning*. Springer International Publishing. https://doi.org/10.1007/978-3-030-93088-2

Datasets - UCI Machine Learning Repository. (2019, September 21). https://archive.ics.uci.edu/datasets?search=Online+Retail&utm_source=chatgpt.com

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis*, *52*(1), 258–271. https://doi.org/10.1016/j.csda.2006.11.025

Lai, H., Huang, T., Lu, B., Zhang, S., und Xiaog, R. (2025). Silhouette coefficient-based weighting k-means algorithm. *Neural Computing and Applications*, *37*(5), 3061–3075. https://doi.org/10.1007/s00521-024-10706-0

Miyabon. (2009–2011, August 31–Dezember 30). *Online Retail II UCI*. https://www.kaggle.com/datasets/mashlyn/online-retail-ii-uci

Moulavi, D., Jaskowiak, P. A., Campello, R. J. G. B., Zimek, A., und Sander, J. (2014). Density-Based Clustering Validation. *Proceedings of the 2014 SIAM International Conference on Data Mining*, 839–847. https://doi.org/10.1137/1.9781611973440.96

Tsiptsis, K., und Chorianopoulos, A. (2010). *Data Mining Techniques in CRM: Inside Customer Segmentation* (1. Aufl.). Wiley. https://doi.org/10.1002/9780470685815

Wirth, R., und Hipp, J. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*. 11. http://cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf

Zykov, R., Noskov, A., und Anokhin, A. (2022). *Retailrocket recommender system dataset*. Kaggle.com. https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset

Anhang

Anhangsverzeichnis

A: CRISP-DM

A.1: Das CRISP-DM-Modell

Die Abbildung A.1 zeigt das CRISP-DM-Referenzmodell mit seinen sechs Phasen Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation und Deployment. Pfeile markieren die zentralen Abhängigkeiten, während der äußere Kreis den iterativen, zyklischen Charakter des Vorgehens unterstreicht. Das Modell ist als allgemeiner Rahmen entworfen und unabhängig von Anwendungsdomäne oder eingesetzter Software.

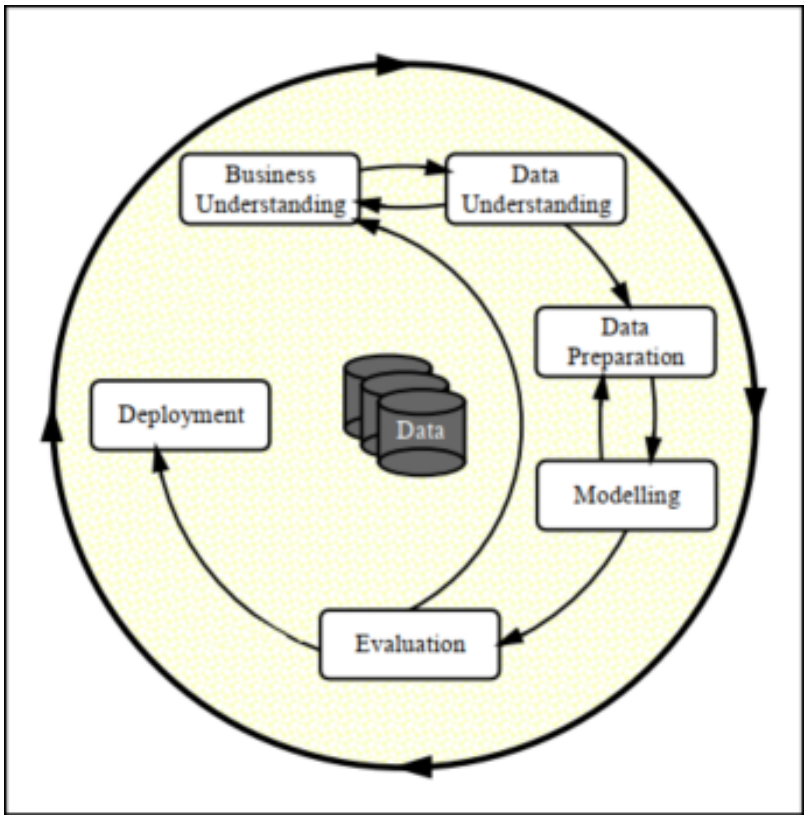

Abbildung A.1: Das CRISP-DM-Modell nach Wirth und Hipp (2000), S. 5

Jede der sechs Phasen ist in spezifische Aufgaben untergliedert, die den Ablauf systematisieren und nachvollziehbar machen. So umfasst Data Understanding das Beschreiben, Erkunden und Überprüfen der Datenqualität, während Data Preparation Aufgaben wie Auswählen, Bereinigen, Konstruieren und Integrieren von Daten vorsieht. Modeling beinhaltet die Wahl geeigneter Verfahren, die Erstellung eines Testdesigns sowie den Aufbau und die Bewertung von Modellen. Evaluation konzentriert sich auf die Überprüfung der Ergebnisse im Hinblick auf die Projektziele, und Deployment behandelt die Bereitstellung und Nutzung der Modelle im Anwendungskontext. Wirth und Hipp betonen außerdem den stark iterativen Charakter des Modells: Ergebnisse einer Phase können jederzeit zu Rücksprüngen führen, wodurch die Arbeitsschritte flexibel angepasst werden können (Wirth und Hipp (2000), S. 6–9).

A.2: Das CRISP-DM-Modell angewendet auf diese Studienarbeit

Die Abbildung A.2 zeigt, wie die einzelnen Phasen des CRISP-DM-Modells konkret auf die Schritte und Kapitel dieser Studienarbeit abgebildet wurden. Die Zuordnung verdeutlicht, welche Arbeitsschritte in welcher Phase umgesetzt und wie sie im Dokument strukturiert sind.

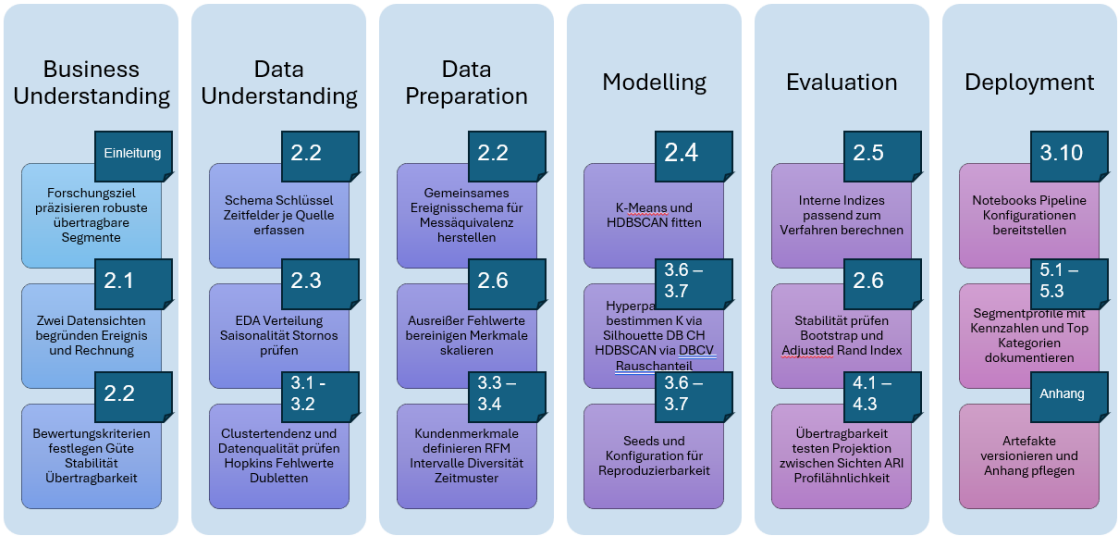

Abbildung A.2: Eigene Darstellung des angewendeten CRISP-DM-Frameworks

Die Grafik ordnet die Kapitel und Arbeitsschritte dieser Studienarbeit den einzelnen Phasen des CRISP-DM-Prozesses zu. So wird sichtbar, wie die theoretischen und praktischen Teile entlang des etablierten Data-Mining-Standards strukturiert und umgesetzt wurden.

B: Kundensegmentierung

B.1: Einordnung der Kundensegmentierung unter Unsupervised Learning

Die Kundensegmentierung ist ein klassisches Beispiel für Clustering und damit ein zentrales Verfahren des unüberwachten Lernens. Im Machine-Learning-Kontext werden Kunden anhand ihrer Merkmale in Gruppen eingeteilt, ohne dass eine Zielvariable vorgegeben ist. Die folgende Abbildung zeigt die Einordnung von Clustering und Kundensegmentierung innerhalb der verschiedenen Typen von Machine Learning.

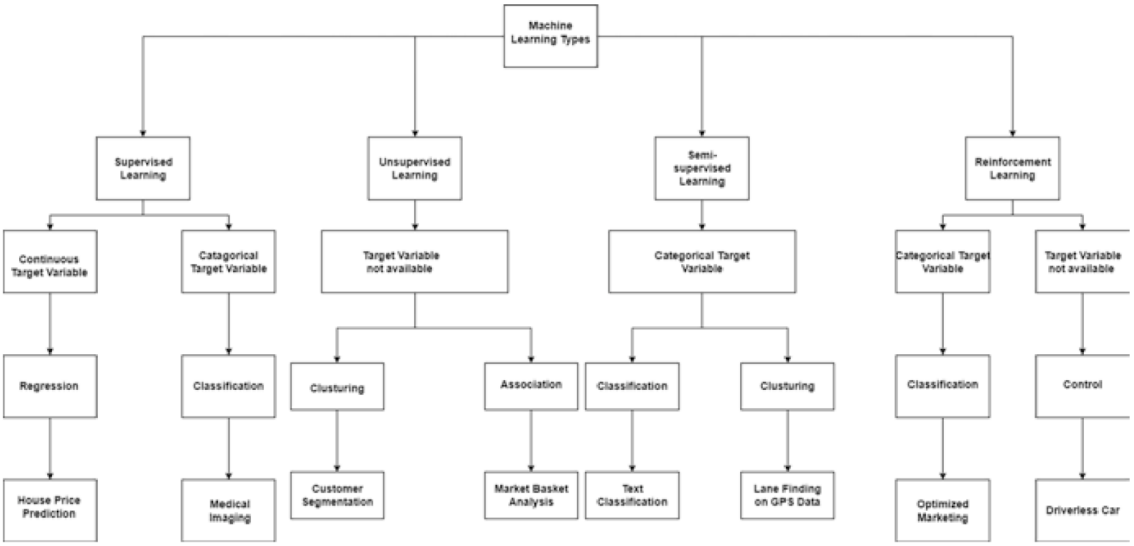

Abbildung A.3: Ansatzpunkte zur Lösung nach Chakraborty et al. (2022), S. 28

Wie in der Abbildung A.3 dargestellt, ist Clustering ein Teilbereich des Unsupervised Learning. Kundensegmentierung nutzt Clustering-Algorithmen, um natürliche Gruppen von Kunden zu identifizieren. Diese Segmentierung bildet die Grundlage für gezielte Marke-

tingmaßnahmen und eine personalisierte Kundenansprache, ohne dass vorher festgelegte Zielvariablen benötigt werden. Im CRM-Kontext ermöglicht dies eine datengetriebene und flexible Einteilung der Kundenbasis.

C: Datenaufbereitung und Voranalysen

Die im Haupttext (vgl. Abschnitt 3.1 Datenbasis, Harmonisierung und Bereinigung) skizzierten Arbeitsschritte werden in diesem Anhang detailliert dokumentiert. Dadurch bleibt der Fließtext der Studienarbeit fokussiert, während Nachvollziehbarkeit und Reproduzierbarkeit gewährleistet sind.

Der Anhang ist in fünf Teile gegliedert:

1. Variablenbeschreibung und Quellen (C.1)
2. Abbildung auf ein einheitliches Ereignisschema (C.2)
3. Bereinigungsschritte (C.3)
4. Normalisierung der Zeitstempel (C.4)
5. Zusammenfassung des bereinigten Datenbestands (C.5)

In den Abschnitten C.1 bis C.4 wird ausschließlich R eingesetzt, um die Daten einzulesen, zu prüfen, zu bereinigen und in ein konsistentes Ereignisschema zu überführen. Mit Paketen wie dplyr, tidyr und here lassen sich die Rohdaten strukturiert verarbeiten und tabellarisch dokumentieren.
Python kommt erst im weiteren Verlauf der Arbeit (ab Kapitel 4, Evaluation und Ergebnisse) zum Einsatz, um die eigentliche Modellierung der Kundensegmente mit Clustering-Algorithmen (K-Means, HDBSCAN) und die Berechnung der Qualitätsmaße umzusetzen.

Diese Aufgabenteilung folgt der Praxis vieler Forschungsarbeiten. R bietet klare Vorteile bei der transparenten Dokumentation und Datenaufbereitung, während Python durch seine umfangreichen Bibliotheken im Machine-Learning-Bereich für die Modellierung prädestiniert ist.

C.1 Variablenbeschreibung und Quellen

```r
# Pakete laden
library(readr)
library(dplyr)
library(here)

# Versionen dokumentieren
pkgs <- c("readr","dplyr","here")
vers <- sapply(pkgs, function(p) as.character(utils::packageVersion(p)))
attached <- pkgs %in% sub("^package:", "", search())
data.frame(package = pkgs, version = vers, attached = attached)
```

Terminal-Output

```
      package version attached
readr   readr   2.1.5     TRUE
dplyr   dplyr   1.1.4     TRUE
here     here   1.0.1     TRUE
```

```r
# Pfade bauen
p_events <- here("data","RR","events.csv")
p_item1  <- here("data","RR","item_properties_part1.csv")
p_item2  <- here("data","RR","item_properties_part2.csv")
p_cat    <- here("data","RR","category_tree.csv")
p_or2    <- if (file.exists(here("data","OR2","online_retail_II.csv"))) {
              here("data","OR2","online_retail_II.csv")
            } else {
              here("data","OR2","online_retail_ll.csv")
            }

# Laden
events        <- read_csv(p_events, show_col_types = FALSE)
item_props_1  <- read_csv(p_item1,  show_col_types = FALSE)
item_props_2  <- read_csv(p_item2,  show_col_types = FALSE)
```

```r
category_tree <- read_csv(p_cat,    show_col_types = FALSE)
online_retail <- read_csv(p_or2,    show_col_types = FALSE)

# Erste 5 Zeilen je Datensatz
head(events, 5)
```

Terminal-Output

```
# A tibble: 5 x 5
      timestamp visitorid event itemid transactionid
          <dbl>     <dbl> <chr>  <dbl>         <dbl>
1 1433221332117    257597 view  355908            NA
2 1433224214164    992329 view  248676            NA
3 1433221999827    111016 view  318965            NA
4 1433221955914    483717 view  253185            NA
5 1433221337106    951259 view  367447            NA
```

```r
head(item_props_1, 5)
```

Terminal-Output

```
# A tibble: 5 x 4
      timestamp itemid property   value
          <dbl>  <dbl> <chr>      <chr>
1 1435460400000 460429 categoryid 1338
2 1441508400000 206783 888        1116713 960601 n277.200
3 1439089200000 395014 400        n552.000 639502 n720.000 424566
4 1431226800000  59481 790        n15360.000
5 1431831600000 156781 917        828513
```

```r
head(item_props_2, 5)
```

Terminal-Output

```
# A tibble: 5 x 4
      timestamp itemid property value
          <dbl>  <dbl> <chr>    <chr>
1 1433041200000 183478 561      769062
2 1439694000000 132256 976      n26.400 1135780
3 1435460400000 420307 921      1149317 1257525
4 1431831600000 403324 917      1204143
5 1435460400000 230701 521      769062
```

```r
head(category_tree, 5)
```

Terminal-Output

```
# A tibble: 5 x 2
  categoryid parentid
       <dbl>    <dbl>
1       1016      213
2        809      169
3        570        9
4       1691      885
5        536     1691
```

```r
head(online_retail, 5)
```

Terminal-Output

```
# A tibble: 5 x 8
  Invoice StockCode Description Quantity InvoiceDate         Price `Customer ID
  <chr>   <chr>     <chr>          <dbl> <dttm>              <dbl>         <dbl
1 489434  85048     "15CM CHRI~       12 2009-12-01 07:45:00  6.95          1308
2 489434  79323P    "PINK CHER~       12 2009-12-01 07:45:00  6.75          1308
3 489434  79323W    "WHITE CHE~       12 2009-12-01 07:45:00  6.75          1308
4 489434  22041     "RECORD FR~       48 2009-12-01 07:45:00  2.1           1308
5 489434  21232     "STRAWBERR~       24 2009-12-01 07:45:00  1.25          1308
# i 1 more variable: Country <chr>
```

Die Ausgaben verdeutlichen die Struktur der Datensätze. Zur besseren Übersicht werden die wichtigsten Variablen und ihre Bedeutungen zusätzlich tabellarisch dargestellt:

Tabelle 5.1: Übersicht der Variablen in beiden Datenquellen

Variable	Quelle	Bedeutung
`timestamp`	RetailRocket (events)	Zeitstempel in Millisekunden seit 1970, Angabe des Ereigniszeitpunkts
`visitorid`	RetailRocket (events)	Anonymisierte Kennung eines Besuchers
`event`	RetailRocket (events)	Typ des Ereignisses: *view*, *addtocart*, *transaction*
`itemid`	RetailRocket (events)	Artikelkennung
`transactionid`	RetailRocket (events)	ID der Transaktion (optional, nur bei Käufen)

Variable	Quelle	Bedeutung
`property`	RetailRocket (item_properties)	Attribut eines Artikels (z. B. Kategorie, Preis, technische Merkmale)
`value`	RetailRocket (item_properties)	Wert des jeweiligen Attributs
`categoryid`	RetailRocket (category_tree)	Identifikator einer Kategorie
`parentid`	RetailRocket (category_tree)	Übergeordnete Kategorie, beschreibt die Hierarchie
`Invoice`	Online Retail II	Rechnungsnummer
`StockCode`	Online Retail II	Artikelcode
`Description`	Online Retail II	Kurzbeschreibung des Artikels
`Quantity`	Online Retail II	Menge der verkauften Artikel
`InvoiceDate`	Online Retail II	Zeitstempel der Rechnungsstellung
`Price`	Online Retail II	Einzelpreis pro Artikel
`Customer ID`	Online Retail II	Anonymisierte Kundenkennung
`Country`	Online Retail II	Land des Kunden

Damit ist nachvollziehbar, dass RetailRocket eine ereignisbasierte Sicht (Nutzung und Interaktion) bereitstellt, während Online Retail II eine transaktionsbasierte Sicht liefert. Beide Perspektiven ergänzen sich und bilden die Grundlage für die in C.2 dargestellte Harmonisierung in ein gemeinsames Ereignisschema.

C.2: Abbildung auf ein einheitliches Ereignisschema

Ziel ist eine gemeinsame Struktur beider Quellen, in der Kunden-, Artikel-, Zeit- und Ereignisinformationen identisch benannt sind. Zusätzliche Felder für Menge, Preis, Umsatz und Kategorie werden bereits angelegt, auch wenn sie in RetailRocket zunächst leer bleiben. Preise und Kategorien werden dort später aus *item_properties* und *category_tree* ergänzt (vgl. C.3 Bereinigungsschritte, C.4 Normalisierung der Zeitstempel).

Zuordnung der Quell- zu den Zielvariablen

Tabelle 5.2: Abbildung der Variablen auf das Ereignisschema

Zielvariable	RetailRocket (Quelle)	Online Retail II (Quelle)	Bedeutung
`customer_id`	`visitorid`	`Customer ID`	Anonymisierte Kennung des Kunden
`item_id`	`itemid`	`StockCode`	Artikelkennung
`timestamp`	`timestamp` (ms seit 1970)	`InvoiceDate`	Zeitpunkt des Ereignisses / der Transaktion
`event_type`	`event` (*view*, *addtocart*, *transaction*)	implizit: immer *purchase*	Typ des Ereignisses
`transactionid`	`transactionid` (falls Kauf)	`Invoice`	Eindeutige Transaktionskennung
`quantity`	nicht enthalten	`Quantity`	Anzahl der verkauften Artikel
`price`	über `item_properties`	`Price`	Einzelpreis pro Artikel
`revenue`	berechnet: quantity × price	berechnet: Quantity × Price	Umsatz pro Ereignis
`category`	aus `category_tree`	ggf. aus `Description`	Zuordnung zu einer Produktkategorie

Während RetailRocket eine feingranulare Unterscheidung nach Ereignistypen erlaubt (*view*, *addtocart*, *transaction*), bildet Online Retail II ausschließlich abgeschlossene Käufe ab. Beide Datensätze liefern somit unterschiedliche, aber komplementäre Informationen.

Die Abbildung in ein gemeinsames Schema stellt sicher, dass spätere Merkmalsberechnungen (z. B. Recency, Frequency, Monetary-Werte, Kaufintervalle, Diversität) auf konsistenter Datenbasis erfolgen können.

Die Umsetzung der Transformation erfolgt mit R. Der folgende Chunk zeigt exemplarisch,

wie die Zielvariablen aus beiden Quellen generiert werden.

```
# RetailRocket auf Ereignisschema abbilden
rr_events <- events %>%
  select(customer_id = visitorid,
         item_id     = itemid,
         timestamp   = timestamp,
         event_type  = event,
         transactionid) %>%
  mutate(timestamp   = as.POSIXct(timestamp/1000, origin="1970-01-01",
tz="UTC"),
         quantity    = ifelse(event_type == "transaction", 1, NA),
         price       = NA_real_,   # später aus item_properties
         revenue     = ifelse(!is.na(quantity) & !is.na(price), quantity *
price, NA_real_),
         category    = NA_character_) # später aus category_tree

# Online Retail II auf Ereignisschema abbilden
or2_events <- online_retail %>%
  select(transactionid = Invoice,
         item_id       = StockCode,
         customer_id   = `Customer ID`,
         timestamp     = InvoiceDate,
         quantity      = Quantity,
         price         = Price,
         country       = Country,
         description   = Description) %>%   # hier mit reinnehmen
  mutate(event_type = "purchase",
         revenue    = quantity * price,
         category   = description)          # hier benutzen


# Vorschau
head(rr_events, 5)
```

Terminal-Output

```
# A tibble: 5 x 9
  customer_id item_id timestamp           event_type transactionid quantity
        <dbl>   <dbl> <dttm>              <chr>              <dbl>    <dbl>
1      257597  355908 2015-06-02 05:02:12 view                  NA       NA
2      992329  248676 2015-06-02 05:50:14 view                  NA       NA
3      111016  318965 2015-06-02 05:13:19 view                  NA       NA
4      483717  253185 2015-06-02 05:12:35 view                  NA       NA
5      951259  367447 2015-06-02 05:02:17 view                  NA       NA
# i 3 more variables: price <dbl>, revenue <dbl>, category <chr>
```

```r
head(or2_events, 5)
```

Terminal-Output

```
# A tibble: 5 x 11
  transactionid item_id customer_id timestamp           quantity price country
  <chr>         <chr>         <dbl> <dttm>                 <dbl> <dbl> <chr>
1 489434        85048         13085 2009-12-01 07:45:00       12  6.95 United K
2 489434        79323P        13085 2009-12-01 07:45:00       12  6.75 United K
3 489434        79323W        13085 2009-12-01 07:45:00       12  6.75 United K
4 489434        22041         13085 2009-12-01 07:45:00       48  2.1  United K
5 489434        21232         13085 2009-12-01 07:45:00       24  1.25 United K
# i 4 more variables: description <chr>, event_type <chr>, revenue <dbl>,
#   category <chr>
```

In der harmonisierten Tabelle von RetailRocket bleiben die Variablen `quantity`, `price`, `revenue` und `category` zunächst leer, da diese Informationen im Rohdatensatz nicht in direkter Form vorliegen. Diese Spalten wurden dennoch angelegt, um die Struktur beider Quellen konsistent zu halten und spätere Ergänzungen zu ermöglichen.

Bei Online Retail II sind diese Variablen hingegen vollständig verfügbar: Transaktionskennungen, Mengen, Preise und Umsätze können direkt übernommen werden. Der Ereignistyp ist dort implizit immer ein Kauf (*purchase*), sodass keine Unterscheidung zwischen verschiedenen Interaktionen möglich ist.

Die Abbildung verdeutlicht damit die komplementäre Natur der Daten: RetailRocket liefert eine feingranulare Sicht auf Nutzerverhalten, Online Retail II eine wertbasierte Sicht auf Käufe. Das vereinheitlichte Ereignisschema stellt sicher, dass in den folgenden Schritten (vgl. C.3 Bereinigungsschritte und C.4 Normalisierung der Zeitstempel) eine konsistente

Datenbasis für die Merkmalsbildung geschaffen wird.

C.3: Bereinigungsschritte

Die Rohdaten enthalten neben den relevanten Informationen auch fehlerhafte oder problematische Einträge. Um eine konsistente Datenbasis für die spätere Merkmalsbildung zu schaffen, wurden die folgenden Regeln angewandt:

1. Pflichtfelder: Beobachtungen ohne Kunden-, Artikel- oder Zeitangabe wurden entfernt.
2. Duplikate: Exakte Dubletten (gleiche Kunden-, Artikel-, Zeit- und Ereignisinformationen) wurden eliminiert.
3. Rückgaben: Im Datensatz Online Retail II wurden negative Mengen sowie Rechnungsnummern mit Präfix „C“ als Rückgaben gekennzeichnet. Diese Ereignisse erhielten den Typ `return`.
4. Ausreißer: Mengen und Preise wurden per Winsorisierung auf das 1. und 99. Perzentil begrenzt, um extreme Werte abzuflachen.
5. Fehlerhafte Werte: Offensichtlich ungültige Beobachtungen (z. B. Mengen 0 nach Winsorisierung) wurden ausgeschlossen.

Im RetailRocket-Datensatz bleiben die Variablen `quantity`, `price` und `revenue` leer, da diese Informationen dort nicht in direkter Form vorliegen. Die Ereignisse konzentrieren sich auf Interaktionen wie *view*, *addtocart* und *transaction*. Im Online-Retail-II-Datensatz konnten dagegen Transaktionskennungen, Mengen, Preise und Umsätze bereinigt und in plausibler Form übernommen werden.

Die Anwendung dieser Regeln stellt sicher, dass Auswertungen auf stabilen Grundlagen erfolgen und Verzerrungen durch fehlerhafte Einträge vermieden werden. Die bereinigten Daten bilden damit die Ausgangsbasis für die Normalisierung der Zeitangaben in Abschnitt C.4.

```r
# benötigte Pakete
library(dplyr)
library(stringr)
```

```r
# Vorbedingungen prüfen
if (!exists("rr_events"))  stop("rr_events fehlt. Bitte C.2 ausführen.")
if (!exists("or2_events")) stop("or2_events fehlt. Bitte C.2 ausführen.")

# Hilfsfunktion: sichere Winsorisierung (tut nichts, wenn zu wenig gültige
Werte vorliegen)
safe_winsorize <- function(x, p_low = 0.01, p_high = 0.99) {
  if (!is.numeric(x)) return(x)
  nn <- sum(!is.na(x))
  if (nn < 10) return(x)
  lo <- suppressWarnings(quantile(x, probs = p_low, na.rm = TRUE))
  hi <- suppressWarnings(quantile(x, probs = p_high, na.rm = TRUE))
  x <- pmin(pmax(x, lo), hi)
  as.numeric(x)
}

# 1) RetailRocket bereinigen
# Pflichtfelder belegen, exakte Duplikate entfernen
rr_clean <- rr_events %>%
  # nur wirklich notwendige Felder
  select(customer_id, item_id, timestamp, event_type, transactionid,
         quantity, price, revenue, category) %>%
  # Pflichtfelder
  filter(!is.na(customer_id), !is.na(item_id), !is.na(timestamp),
!is.na(event_type)) %>%
  # exakte Duplikate
  distinct(customer_id, item_id, timestamp, event_type, transactionid,
.keep_all = TRUE)

# 2) Online Retail II bereinigen
# Rückgaben-Regel und Felder füllen
or2_clean <- or2_events %>%
  select(transactionid, item_id, customer_id, timestamp, quantity, price,
country, description,
```

```
         event_type, revenue, category) %>%
  mutate(
    quantity = suppressWarnings(as.numeric(quantity)),
    price    = suppressWarnings(as.numeric(price))
  ) %>%
  filter(!is.na(customer_id), !is.na(item_id), !is.na(timestamp)) %>%
  mutate(
    return_flag  = (!is.na(quantity) & quantity < 0) |
                   (!is.na(transactionid) & grepl("^C",
as.character(transactionid))),
    quantity_abs = if_else(return_flag & !is.na(quantity), abs(quantity),
quantity),
    event_type   = if_else(return_flag, "return", "purchase"),
    category     = coalesce(category, description),
    revenue      = if_else(!is.na(quantity_abs) & !is.na(price), quantity_abs
* price, revenue)
  ) %>%
  distinct(transactionid, item_id, customer_id, timestamp, quantity_abs,
price, event_type, .keep_all = TRUE) %>%
  mutate(
    quantity_w = safe_winsorize(quantity_abs, 0.01, 0.99),
    price_w    = safe_winsorize(price,       0.01, 0.99),
    revenue_w  = quantity_w * price_w
  ) %>%
  filter(is.na(quantity_w) | quantity_w > 0,
         is.na(price_w)    | price_w    > 0)


# kurze Sichtprüfungen
head(rr_clean, 5)
```

Terminal-Output

```
# A tibble: 5 x 9
  customer_id item_id timestamp           event_type transactionid quantity
        <dbl>   <dbl> <dttm>              <chr>              <dbl>    <dbl>
1      257597  355908 2015-06-02 05:02:12 view                  NA       NA
2      992329  248676 2015-06-02 05:50:14 view                  NA       NA
3      111016  318965 2015-06-02 05:13:19 view                  NA       NA
4      483717  253185 2015-06-02 05:12:35 view                  NA       NA
5      951259  367447 2015-06-02 05:02:17 view                  NA       NA
# i 3 more variables: price <dbl>, revenue <dbl>, category <chr>
```

```r
head(or2_clean, 5)
```

Terminal-Output

```
# A tibble: 5 x 16
  transactionid item_id customer_id timestamp           quantity price country
  <chr>         <chr>         <dbl> <dttm>                 <dbl> <dbl> <chr>
1 489434        85048         13085 2009-12-01 07:45:00       12  6.95 United K
2 489434        79323P        13085 2009-12-01 07:45:00       12  6.75 United K
3 489434        79323W        13085 2009-12-01 07:45:00       12  6.75 United K
4 489434        22041         13085 2009-12-01 07:45:00       48  2.1  United K
5 489434        21232         13085 2009-12-01 07:45:00       24  1.25 United K
# i 9 more variables: description <chr>, event_type <chr>, revenue <dbl>,
#   category <chr>, return_flag <lgl>, quantity_abs <dbl>, quantity_w <dbl>,
#   price_w <dbl>, revenue_w <dbl>
```

Die Vorschau zeigt, dass Pflichtfelder belegt sind, Duplikate entfernt wurden und Rückgaben in Online Retail II über `event_type = return` gekennzeichnet sind. Mengen und Preise wurden dort per Winsorisierung begrenzt und als `quantity_w` und `price_w` bereitgestellt.

C.4: Normalisierung der Zeitstempel

Neben inhaltlichen Bereinigungen ist auch die einheitliche Behandlung der Zeitangaben erforderlich. Beide Datensätze enthalten Zeitinformationen, die jedoch in unterschiedlicher Form vorliegen.

- Im RetailRocket-Datensatz werden Zeitstempel in Millisekunden seit dem 1. Januar 1970 gespeichert
- Im Online-Retail-II-Datensatz liegt das Kaufdatum bereits als Datumsfeld (Invoice-

Date) vor, allerdings ohne einheitliche Ableitung von Kalendermerkmalen

Um die Daten konsistent nutzen zu können, wurden folgende Schritte durchgeführt:

1. Konvertierung in ein einheitliches Format. Alle Zeitangaben wurden in das POSIXct-Format überführt und zusätzlich als Datum (date) abgespeichert.
2. Ableitung von Kalendermerkmalen. Aus den Zeitstempeln wurden standardisierte Variablen für Wochentag (wday), Stunde (hour), Monat (month) und Jahr (year) berechnet.
3. Harmonisierung der Skalen. Damit liegen beide Quellen auf derselben Zeitskala vor, was aggregierte Analysen und Vergleiche zwischen RetailRocket und Online Retail II ermöglicht.

Die Normalisierung erlaubt es, zeitliche Muster in beiden Quellen konsistent zu erfassen, beispielsweise bevorzugte Kaufzeiten oder saisonale Effekte. Diese Variablen werden in der Merkmalsbildung (vgl. Abschnitt 3.2) eingesetzt, um Kundencluster auch anhand von Verhaltensrhythmen zu unterscheiden.

```
# Benötigte Pakete laden
library(lubridate)
library(dplyr)

# Guard: C.3 muss gelaufen sein
if (!exists("rr_clean") || !exists("or2_clean")) {
  stop("C.4 benötigt rr_clean und or2_clean aus C.3. Bitte zuerst C.3
ausführen oder das Dokument vollständig rendern.")
}

# RetailRocket
# timestamp ist bereits POSIXct (C.2). Hier werden Datum und Kalendermerkmale
ergänzt
rr_time <- rr_clean %>%
  mutate(
    # reines Datum zusätzlich ablegen
    date  = as.Date(timestamp),
```

```r
    # Wochentag Mo-So, abgekürzt, Montag als Wochenbeginn
    wday  = wday(timestamp, label = TRUE, abbr = TRUE, week_start = 1),
    # Stunde 0-23
    hour  = hour(timestamp),
    # Monat Jan-Dez, abgekürzt
    month = month(timestamp, label = TRUE, abbr = TRUE),
    # Jahr vierstellig
    year  = year(timestamp)
  )

# Online Retail II
# gleiches Verfahren für Konsistenz der Skalen und Merkmale
or2_time <- or2_clean %>%
  mutate(
    date  = as.Date(timestamp),
    wday  = wday(timestamp, label = TRUE, abbr = TRUE, week_start = 1),
    hour  = hour(timestamp),
    month = month(timestamp, label = TRUE, abbr = TRUE),
    year  = year(timestamp)
  )

# Stichproben zur Kontrolle
head(rr_time, 5)
```

Terminal-Output

```
# A tibble: 5 x 14
  customer_id item_id timestamp           event_type transactionid quantity
        <dbl>   <dbl> <dttm>              <chr>              <dbl>    <dbl>
1      257597  355908 2015-06-02 05:02:12 view                  NA       NA
2      992329  248676 2015-06-02 05:50:14 view                  NA       NA
3      111016  318965 2015-06-02 05:13:19 view                  NA       NA
4      483717  253185 2015-06-02 05:12:35 view                  NA       NA
5      951259  367447 2015-06-02 05:02:17 view                  NA       NA
# i 8 more variables: price <dbl>, revenue <dbl>, category <chr>, date <date>,
#   wday <ord>, hour <int>, month <ord>, year <dbl>
```

```r
head(or2_time, 5)
```

Terminal-Output

```
# A tibble: 5 x 21
  transactionid item_id customer_id timestamp           quantity price country
  <chr>         <chr>         <dbl> <dttm>                 <dbl> <dbl> <chr>
1 489434        85048         13085 2009-12-01 07:45:00       12  6.95 United K
2 489434        79323P        13085 2009-12-01 07:45:00       12  6.75 United K
3 489434        79323W        13085 2009-12-01 07:45:00       12  6.75 United K
4 489434        22041         13085 2009-12-01 07:45:00       48  2.1  United K
5 489434        21232         13085 2009-12-01 07:45:00       24  1.25 United K
# i 14 more variables: description <chr>, event_type <chr>, revenue <dbl>,
#   category <chr>, return_flag <lgl>, quantity_abs <dbl>, quantity_w <dbl>,
#   price_w <dbl>, revenue_w <dbl>, date <date>, wday <ord>, hour <int>,
#   month <ord>, year <dbl>
```

Die Normalisierung liegt damit für beide Datensätze in identischer Struktur vor. Die abgeleiteten Kalendermerkmale werden in der Merkmalsbildung genutzt und erlauben zeitliche Musteranalysen ohne zusätzliche Transformationen.

C.5: Zusammenfassung des bereinigten Datenbestands

Die Abschnitte C.3 und C.4 haben die Bereinigung und die Normalisierung der Zeitangaben dokumentiert. In diesem Abschnitt wird der bereinigte Datenbestand in komprimierter Form zusammengefasst. Dadurch lässt sich die Qualität der Datenbasis einschätzen, die in Abschnitt 3.2 Merkmalsbildung für die Ableitung der Cluster-Merkmale genutzt wird.

Zentrale Fragen der Plausibilisierung sind:

1. Wie groß sind die Datensätze nach der Bereinigung?
2. Welche Verteilungen zeigen die kaufbezogenen Variablen in Online Retail II?
3. Sind die Strukturen der beiden Quellen konsistent nutzbar?

```r
library(dplyr)


# Dimensionen beider Quellen
dim_summary <- data.frame(
```

```
  datensatz  = c("RetailRocket (bereinigt)", "Online Retail II (bereinigt)"),
  n_zeilen   = c(nrow(rr_time), nrow(or2_time)),
  n_spalten  = c(ncol(rr_time), ncol(or2_time))
)

# Verteilungen kaufbezogener Variablen in Online Retail II
or2_summary <- or2_time %>%
  summarise(
    n_events       = n(),
    n_returns      = sum(event_type == "return", na.rm = TRUE),
    median_qty     = median(quantity_w, na.rm = TRUE),
    p99_qty        = quantile(quantity_w, 0.99, na.rm = TRUE),
    median_price   = median(price_w, na.rm = TRUE),
    p99_price      = quantile(price_w, 0.99, na.rm = TRUE),
    median_revenue = median(revenue_w, na.rm = TRUE),
    p99_revenue    = quantile(revenue_w, 0.99, na.rm = TRUE)
  )

# Stichprobenzeilen zur Illustration
rr_probe  <- head(select(rr_time, customer_id, item_id, timestamp,
event_type), 5)
or2_probe <- head(select(or2_time, customer_id, item_id, timestamp,
event_type,
                         quantity_w, price_w, revenue_w), 5)

dim_summary
```

Terminal-Output

```
                   datensatz n_zeilen n_spalten
1     RetailRocket (bereinigt)  2755641        14
2 Online Retail II (bereinigt)   797883        21
```

```
or2_summary
```

Terminal-Output

```
# A tibble: 1 x 8
  n_events n_returns median_qty p99_qty median_price p99_price median_revenue
     <int>     <int>      <dbl>   <dbl>        <dbl>     <dbl>          <dbl>
1   797883     18390          5     144         1.95      15.0           12.5
# i 1 more variable: p99_revenue <dbl>
```

```
rr_probe
```

Terminal-Output

```
# A tibble: 5 x 4
  customer_id item_id timestamp           event_type
        <dbl>   <dbl> <dttm>              <chr>
1      257597  355908 2015-06-02 05:02:12 view
2      992329  248676 2015-06-02 05:50:14 view
3      111016  318965 2015-06-02 05:13:19 view
4      483717  253185 2015-06-02 05:12:35 view
5      951259  367447 2015-06-02 05:02:17 view
```

```
or2_probe
```

Terminal-Output

```
# A tibble: 5 x 7
  customer_id item_id timestamp           event_type quantity_w price_w
        <dbl> <chr>   <dttm>              <chr>           <dbl>   <dbl>
1       13085 85048   2009-12-01 07:45:00 purchase           12    6.95
2       13085 79323P  2009-12-01 07:45:00 purchase           12    6.75
3       13085 79323W  2009-12-01 07:45:00 purchase           12    6.75
4       13085 22041   2009-12-01 07:45:00 purchase           48    2.1
5       13085 21232   2009-12-01 07:45:00 purchase           24    1.25
# i 1 more variable: revenue_w <dbl>
```

Die Übersicht zeigt, dass beide Quellen nach der Bereinigung eine substanzielle Anzahl an Beobachtungen enthalten und somit ausreichend groß für die geplante Clusteranalyse sind.

Die kaufbezogenen Variablen im Online-Retail-II-Datensatz weisen nach Winsorisierung plausible Wertebereiche auf, extreme Ausreißer wurden erfolgreich begrenzt. Rückgaben sind als eigene Ereignisse gekennzeichnet und können später gesondert berücksichtigt werden.

Der RetailRocket-Datensatz bildet dagegen ausschließlich Interaktionen ab, liefert aber mit den normalisierten Zeitfeldern eine konsistente Ergänzung zur transaktionsorientierten Sicht von Online Retail II.

Damit ist eine robuste und vergleichbare Datenbasis geschaffen, auf der in Abschnitt 3.2 Merkmalsbildung die eigentlichen Analysemerkmale konstruiert werden.

Glossar

Multi-Faktor-Authentifizierung (MFA) eine Sicherheitsmethode, die eine zusätzliche Verifikationsebene erfordert, beispielsweise durch eine Kombination aus Passwort und biometrischer Authentifizierung.

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Studienarbeitselbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Bielefeld, den 08. September 2025

Christian Roth