



Sprich mit den Daten: Datenanalyse mit ChatGPT im Praxistest

Fabian Heidenstecker, Opitz Consulting

Die Analyse strukturierter Daten gehört heute zum Alltag vieler IT-Abteilungen. Ob Kundenverhalten, Vertriebserfolge oder Betriebskennzahlen – überall entstehen Daten, die es zu verstehen und zu nutzen gilt. Doch klassische Werkzeuge wie SQL, Excel oder Business-Intelligence-Tools setzen meist technisches Know-how voraus. Generative KI, etwa in Form von ChatGPT, verspricht eine neue Form der Interaktion: Natürlichsprachliche Abfragen, automatische Visualisierungen, sogar erste Machine-Learning-Modelle – direkt im Chatfenster. Doch wie gut funktioniert das in der Praxis? In diesem Erfahrungsbericht zeige ich Schritt für Schritt, wie ich mit ChatGPT ein vollständiges Data-Analytics-Projekt durchgeführt habe – ohne eigene Entwicklungsumgebung, allein mit Prompts und einem realitätsnahen Unternehmensdatensatz.

Wie leistungsfähig ist die GenAI-basierte Datenanalyse?

Traditionell war tiefgreifende Datenanalyse Expertinnen und Experten vorbehalten – mit spezialisierten Tools, komplexer Syntax und oft steiler Lernkurve. Doch mit der Verfügbarkeit leistungsfähiger Sprachmodelle wie ChatGPT verändert sich dieser Paradigmenwechsel: Strukturierte Daten können heute dialogbasiert analysiert, visualisiert und interpretiert werden – ohne tiefgreifende Programmierkenntnisse oder dedizierte Software. Aber wie gut und verlässlich sind diese Modelle?

In diesem Projekt demonstriere ich anhand eines praxisnahen Szenarios, wie sich eine Datenanalyse durch GenAI anfühlt und ob die Ergebnisse wirklich halten, was sie versprechen. Die Basis bildet ein synthetischer Unternehmensdatensatz mit Informationen zu Kunden, Produktkategorien und Kaufverhalten – ein typisches Setup, wie es in vielen Unternehmen existiert.

```

Analyse

python          Details immer anzeigen ⚡ Kopieren

# Das Dataset ist durch Semikolon getrennt, nicht Tab
df = pd.read_csv(file_path, sep=";")

# Vorschau anzeigen
import ace_tools as tools; tools.display_dataframe_to_user(name)

STDOUT/STDERR
/home/sandbox/.local/lib/python3.11/site-packages/pandas/core/
values = values.astype(str)

Ergebnis
ID Year_Birth Education Marital_Status Income Kid
0 5524 1957 Graduation Single 58138.0
1 2174 1954 Graduation Single 46344.0
2 4141 1965 Graduation Together 71613.0
3 6182 1984 Graduation Together 26646.0
4 5324 1981 PhD Married 58293.0

Dt_Customer Recency MntWines ... NumWebVisitsMonth Ac
0 2012-09-04 58 635 ... 7
1 2014-03-08 38 11 ... 5
2 2013-08-21 26 426 ... 4
3 2014-02-10 26 11 ... 6
4 2014-01-19 94 173 ... 5

```

Abbildung 1: Dateianalyse mit generierten Kommentaren (Quelle: Fabian Heidenstecker)

Meine Zielsetzung

Ziel meiner Analyse ist es, zwei zentrale Fragen zu beantworten:

1. Wie leistungsfähig sind GenAI-Tools wie ChatGPT bei typischen Aufgaben der Datenanalyse? Dazu gehören das Laden, Transformieren, Visualisieren und Interpretieren strukturierter Daten.
2. Wie gut lassen sich datenbasierte Erkenntnisse in kontextualisierte Aussagen überführen – etwa in Form von Data-Driven Personas oder einfachen Vorhersagemodellen?

Die Analyse erfolgte vollständig im Dialog mit ChatGPT – ohne IDE, rein über Prompts im Chatfenster.

Hinweis: Die im Artikel abgebildeten Prompts wurden teilweise zur besseren Lesbarkeit gekürzt. Die vollständigen Versionen sind über die Quellangaben verlinkt.

Datengrundlage:
Marketingkampagnen und Kundendaten

Der verwendete Datensatz enthält Informationen zu über 2.000 Kundinnen und Kunden – von soziodemografischen Angaben wie Alter, Bildungsgrad oder Einkommen bis hin zu Kaufverhalten über verschiedene Vertriebskanäle. Enthalten sind unter anderem Ausgaben für Waren-

gruppen wie Wein, Fleisch oder Gold, Reaktionen auf frühere Kampagnen und genutzte Vertriebskanäle wie Online, Filiale oder Katalog. Die Struktur ist typisch für CRM-Systeme oder Data Warehouses im Retail-Umfeld – damit ein ideales Testfeld für KI-gestützte Datenanalyse.

Die Daten spiegeln typische Strukturen realer Vertriebs- und Marketingdaten wider – ideal, um das Potenzial generativer KI zu demonstrieren.

Daten einlesen mit ChatGPT

Über die Dateiupload-Funktion in ChatGPT (GPT-4o mit Advanced Data Analysis)



Abbildung 2: Boxplots der Ausreißer (Quelle: Fabian Heidenstecker)

Ausgaben Und Kundenanzahl Je Kohorte				
Kohorte	Anzahl Kunden	Ø Gesamtausgabe (Mnt*)	Gesamtausgabe (Mnt*)	
1 1940	107	933.6448598130842	99900	
2 1950	460	673.6195652173913	309865	
3 1960	506	623.4110671936759	315446	
4 1970	740	523.222972972973	387185	
5 1980	363	532.2506887052342	193207	
6 1990	61	810.5737704918033	49445	

Abbildung 3: Erste Kennzahlen nach Kohorte (Quelle: Fabian Heidenstecker)

ließ sich die CSV-Datei direkt einlesen – inklusive automatischer Erkennung des Semikolon-Trennzeichens (*siehe Abbildung 1*).

Schon nach dem Laden machte ChatGPT Vorschläge für mögliche nächste Schritte. Ich entschied mich zunächst für eine Prüfung der Datenqualität. Hier überzeugte mich vor allem die automatische tabellarische Darstellung mit Hinweisen auf fehlende Werte und potenzielle Ausreißer – ganz ohne manuelle Rückfragen.

Die Ausreißer der Geburtsjahre ließ ich mir als Boxplot darstellen (*siehe Abbildung 2*).

Erste Kennzahlen und Gruppenanalysen

Mit einfachen Prompts wie „Wie viele Kunden enthält der Datensatz und wie hoch ist das durchschnittliche Einkommen?“ und „Wie viel gibt ein alleinstehender Kunde durchschnittlich für Wein aus?“ lieferte ChatGPT schnell Ergebnisse. Auch gruppierte Analysen waren möglich, etwa:

„Bitte gruppieren die Weinausgaben nach Alterskohorten (10-Jahres-Schritte nach Geburtsjahr).“

Die Resultate wirkten plausibel – und hielten auch dem Excel-Gegencheck stand (dort habe ich die Werte nachgerechnet, was sich für den überschaubaren Datensatz angeboten hat).

Ermutigt von den Ergebnissen, stellte ich fortgeschrittenere Fragen, zum Beispiel: „Wie hoch sind die Ausgaben für Wein, gruppiert nach dem Geburtsjahr?“ Außerdem ließ ich mir die Geburtsjahre in 10-Jahres-Schritten gruppieren, also in Alterskohorten (*siehe Abbildung 3*). Auch hier war ich mit dem Ergebnis zufrieden.

Wichtig zu verstehen

ChatGPT liefert auf Rechenfragen häufig korrekte Antworten, was den Eindruck erweckt, es könne rechnen. Tatsächlich basiert die Ausgabe jedoch auf Wahrscheinlichkeiten, die aus Sprachmustern im Training abgeleitet wurden. Als Sprachmodell führt es keine echten Berechnungen durch, sondern erzeugt bei Bedarf Code, der von einer extra Rechenumgebung ausgeführt werden kann. Die Fähigkeit zur scheinbaren Berechnung ist somit eine Folge der Mustererkennung, nicht mathematischer Kompetenz.

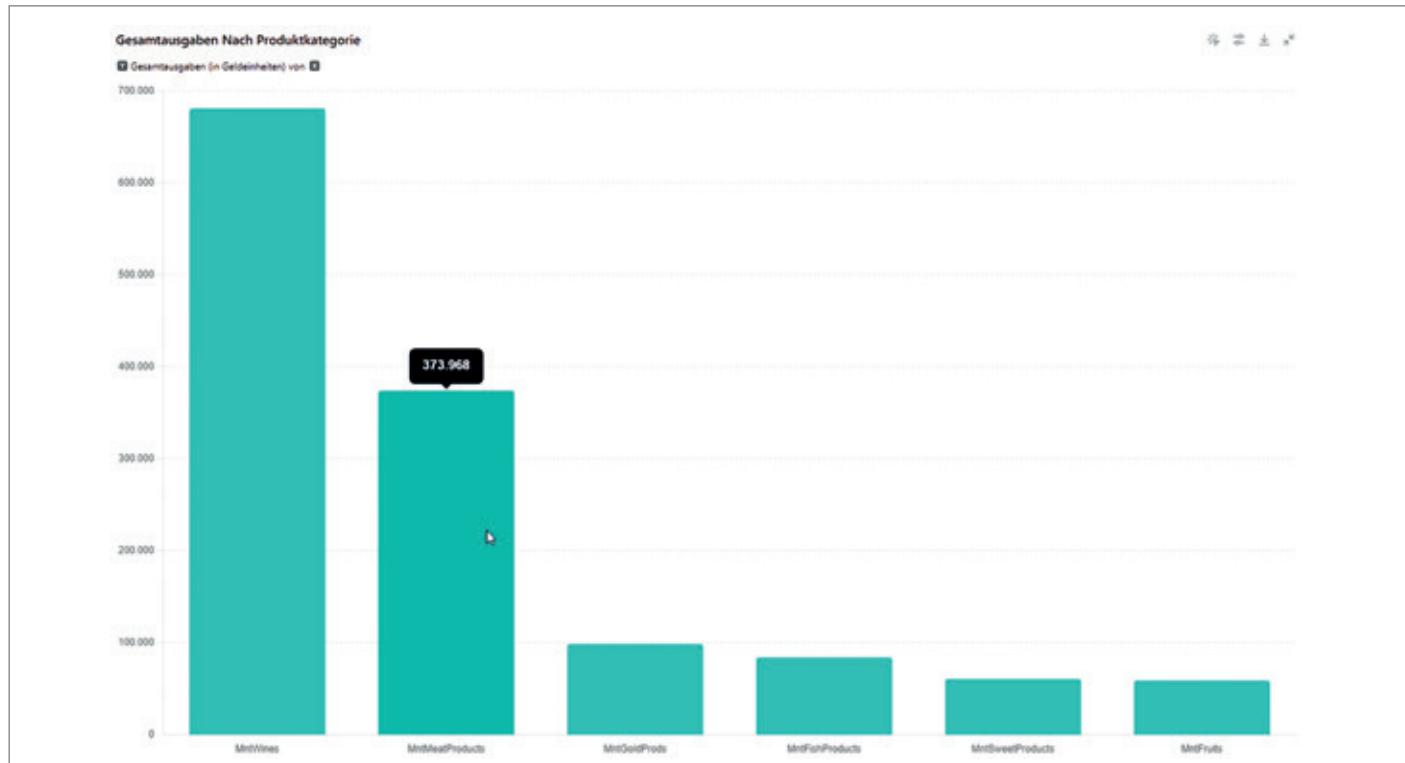


Abbildung 4 Interaktives Säulendiagramm (Quelle: Fabian Heidenstecker)

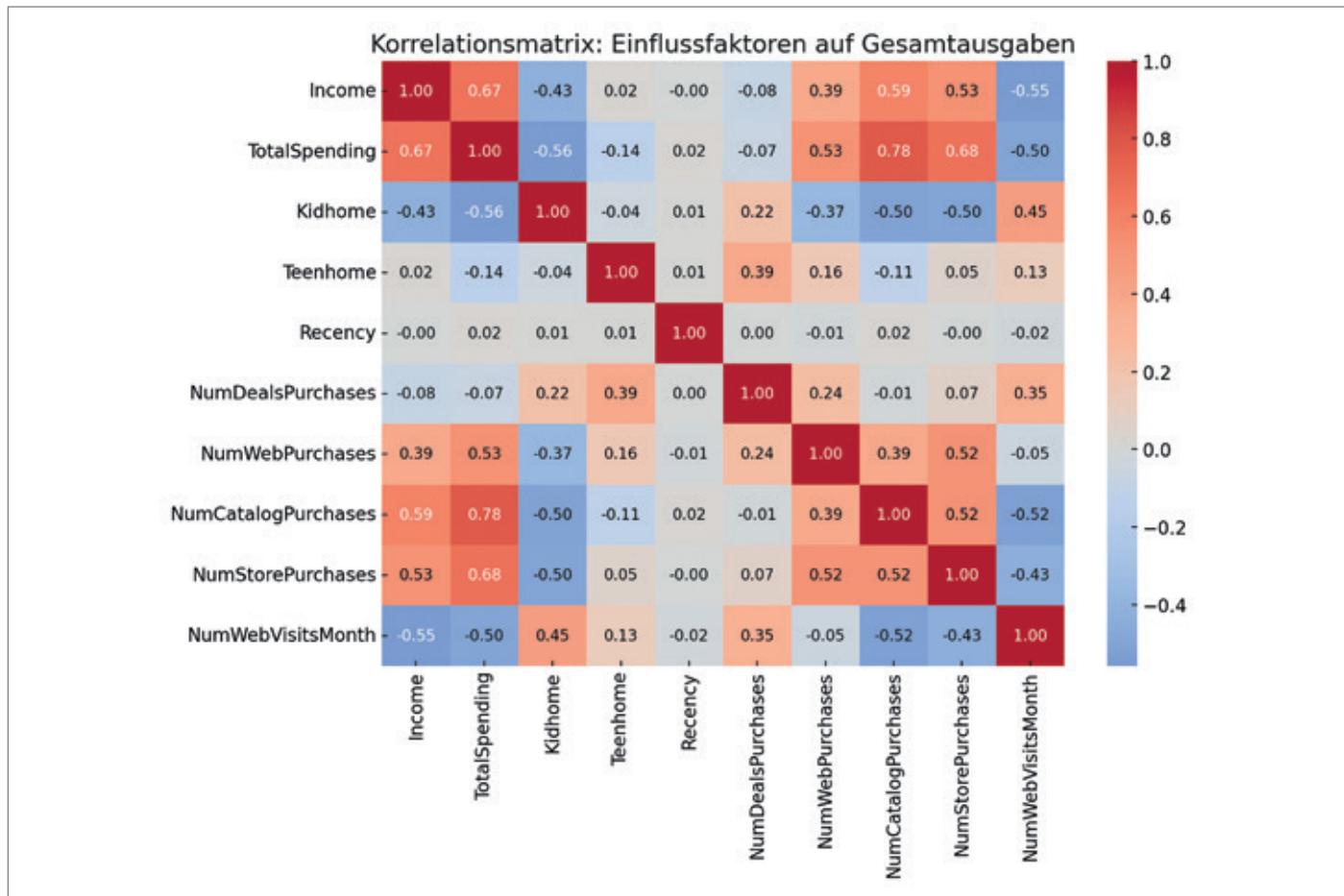


Abbildung 5: Korrelationsmatrix (Quelle: Fabian Heidenstecker)

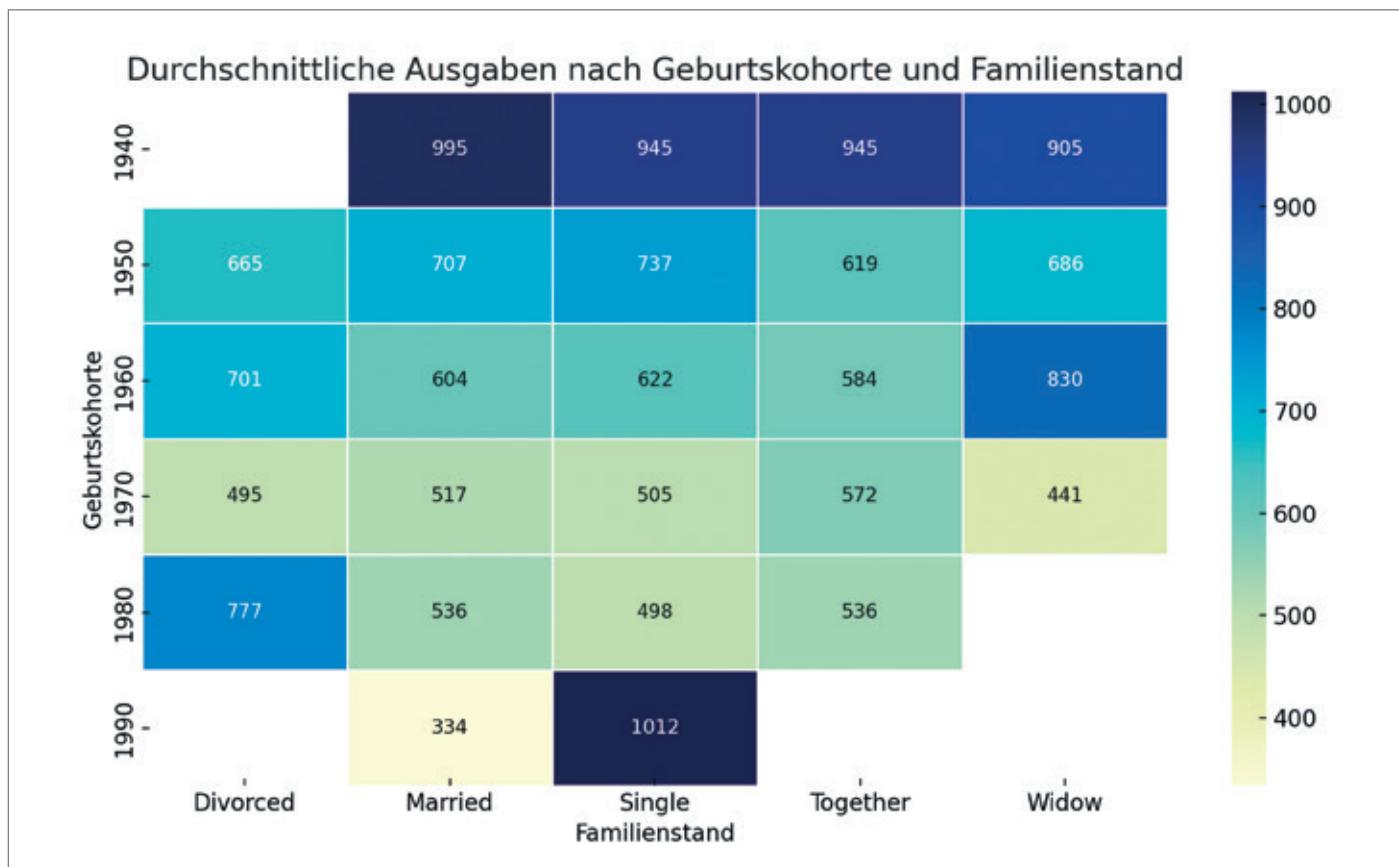


Abbildung 6: Heatmap (Quelle: Fabian Heidenstecker)

Bildung	Familienstand	Einkommen	Kinder (klein/Teen)	Geburtsjahr	Beschwert	Prognose Ausgaben
Graduation	Married	50.000 €	1 / 1	1980	Nein	251.70
PhD	Together	75.000 €	0 / 0	1970	Nein	1.507,24
Master	Single	30.000 €	2 / 0	1990	Ja	108,75

Abbildung 7: Prognose-Tabelle (Quelle: Fabian Heidenstecker)

```

❸ modellvergleich.py > ...
54
55 # -----
56 # 4. Evaluation pro Modell
57 # -----
58 results = []
59
60 for name, model in models.items():
61     print(f"Trainiere Modell: {name}")
62
63     pipeline = Pipeline(steps=[
64         ("preprocessor", ColumnTransformer(
65             transformers=[("cat", OneHotEncoder(drop="first"), cat_features)],
66             remainder="passthrough"
67         )), 
68         ("regressor", model)
69     ])
70
71     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)
72
73     pipeline.fit(X_train, y_train)
74     y_pred = pipeline.predict(X_test)
75
76     mae = mean_absolute_error(y_test, y_pred)
77     rmse = np.sqrt(mean_squared_error(y_test, y_pred))
78
79     results.append({
80         "Modell": name,
81         "MAE": round(mae, 2),
82         "RMSE": round(rmse, 2)
83     })
84
85 # -----
86 # 5. Ergebnisse anzeigen
87 # -----
88 results_df = pd.DataFrame(results).sort_values("MAE")
89 print("\nModellvergleich (nach MAE sortiert):\n")
90 print(results_df.to_string(index=False))
91

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Trainiere Modell: Linear Regression
Trainiere Modell: Random Forest
Trainiere Modell: Random Forest
Trainiere Modell: Gradient Boosting
Trainiere Modell: XGBoost

Modellvergleich (nach MAE sortiert):

Modell	MAE	RMSE
XGBoost	201.14	307.44
Random Forest	201.32	300.09
Gradient Boosting	212.61	309.82
Linear Regression	293.33	391.15

Abbildung 8: Python-Code für lokale Ausführung (Quelle: Fabian Heidenstecker)

Visualisierungen direkt im Chat

Auch die Visualisierung funktionierte erstaunlich gut. Ob Säulendiagramm, Heatmap oder Boxplot – die Diagramme wurden automatisch generiert und waren direkt im Chatfenster anklickbar und exportierbar.

Beispiel „Säulendiagramm“:

Prompt: „Erzeuge ein Säulendiagramm, das die Verkaufsvolumina der Warengruppen zeigt.“

Ergebnis: Die Visualisierung war ansprechend, aber optisch musste ich da noch mal dran. Was mir leicht gemacht wurde, denn direkt auf der Oberfläche konnte ich Details nachjustieren – etwa Farbgebung oder Skalierung (*siehe Abbildung 4*). Besonders hilfreich: Der interaktive Modus mit Tooltips, die präzise Werte beim Mouse-over anzeigen.

Als nächstes wollte ich herausfinden, welches grundlegende Verständnis ChatGPT von der Visualisierung hat und bat um Vorschläge zur Darstellung des Merkmals „Familienstand“. ChatGPT lieferte zwei Optionen – Balkendiagramm für absolute, Kreisdiagramm für relative Häufigkeiten – und verwies explizit auf Vor- und Nachteile. Fachlich fundiert, stilistisch sauber.

Beispiel „Boxplot“

Prompt: Um eine fortgeschrittene Visualisierungsmethode zu testen, bat ich ChatGPT, einen Boxplot zu erzeugen. Boxplots sind in der Statistik beliebt, um die Verteilung von Werten zu zeigen. Dieser Diagrammtyp ist nicht so gängig und auch nicht in allen Visualisierungstools (wie zum Beispiel Power BI) als Standard verfügbar.

Ergebnis: Auch hier lieferte das Modell ein solides Ergebnis, inklusive Interpretation: Höherer Bildungsgrad korreliert mit höherem Einkommen. Ausreißer werden auf Wunsch entfernt.

Korrelationen und Zusammenhänge

Ein Highlight war für mich die automatische Erstellung einer Korrelationsmatrix – inklusive Interpretation. So wurde beispielsweise der erwartete Zusammenhang zwischen Einkommen und Ausgaben direkt erkannt, erläutert und grafisch aufbereitet. Hier zwei Beispiele:

• **Korrelationsmatrix erstellen**

Prompt: „Erstelle eine Korrelationsmatrix aller numerischen Variablen und interprete die stärksten Zusammenhänge.“

Ergebnis: ChatGPT lieferte nicht nur die Matrix selbst, sondern auch gleich eine textuelle Auswertung – etwa den erwartbaren Zusammenhang zwischen Einkommen und Ausgaben oder zwischen Bildung und Kaufverhalten (*siehe Abbildung 5*).

• **Zusammenhang zwischen kategorischen Merkmalen**

In der weiter oben beschriebenen Heatmap hatte ChatGPT visualisiert, wie Alter (Geburtsjahr) und Familienstand der Personen aus meinem Datenset zusammenhängen – mit Farbcodierung nach durchschnittlichen Ausgaben. Besonders hilfreich: Die Gruppengrößen ließen sich direkt als Zusatzinformation in Klammern darstellen.

Prompt: „Zeige den Zusammenhang zwischen den Geburtsjahren und dem Familienstand, bezogen auf die Gesamt-Ausgaben. Ich hätte gerne eine Heatmap.“

Ergebnis: Auch hier erzeugte ChatGPT die passende Visualisierung samt Kontextualisierung in natürlicher Sprache (*siehe Abbildung 6*).

Vorhersagemodell mit Machine Learning

Eine Frage, über die aktuell noch viel spekuliert wird: Wie gut beherrscht ChatGPT Machine-Learning-Algorithmen, beziehungsweise ist es in der Lage, ein Vorhersagemodell zu erstellen?

Prompt: Erstelle ein Vorhersagemodell. Ich möchte verschiedene demographische Attribute und das Einkommen eingeben und eine Vorhersage der Gesamtausgaben für alle Warengruppen erhalten (*siehe Abbildung 7*).

Ergebnis: Heraus kam ein vollständiges Modell inklusive Berechnung des Mean Absolute Error (~201 €) – durchaus solide für einen ersten Wurf. Es wurde ein Random-Forrest-Algorithmus gewählt. Die Prognose erfolgte direkt im Chat – ohne externe IDE. Auch das Ausführen des Modells mit zufälligen Testdaten war direkt im Chat machbar und das Ergebnis wurde als Tabelle dargestellt, die sich per Knopfdruck auch noch exportieren ließ.

An dieser Stelle wich ich von meiner ursprünglichen Zielstellung ab. Ich woll-

te nämlich herausfinden, ob der Code für die Vorhersage auch lokal ausgeführt werden kann. Denn für einen realen Anwendungsfall würde man die Werte beispielsweise in einem Nachlauf berechnen lassen, um sie dann zum Beispiel in das DWH zu integrieren und gegebenenfalls in einem Report anzeigen zu lassen. Den Code übernahm ich per Copy & Paste in eine Python-Umgebung. Als kleinen Service generierte ChatGPT den passenden Befehl, um die notwendigen Python-Bibliotheken zu installieren (*siehe Abbildung 8*).

Der circa 100 Zeilen lange Code ließ sich problemlos ausführen, es gab keine Fehlermeldungen oder Warnungen. ChatGPT lieferte einen gut strukturierten Code, auch Kommentare waren enthalten. Der Code war syntaktisch und strukturell sauber – ideal für produktionsnahe Tests.

Um hier noch mehr Eindrücke zu gewinnen und zu prüfen, ob die Vorhersagequalität noch besser werden kann, habe ich den Code um eine Schleife erweitern lassen. Diese trainierte nacheinander verschiedene Modelle auf Basis bekannter Algorithmen wie etwa XGBoost, und gab die entsprechenden Qualitätskennzahlen aus.

Ergebnis: Insgesamt war ich aber mehr als beeindruckt. Mit nur ein paar Prompts erstellte man Vorhersagemodelle in Python. Natürlich habe ich ein Grundverständnis von den Algorithmen, weiß was Python-Bibliotheken sind, und kann mir lokal eine virtuelle Umgebung einrichten. Der springende Punkt ist, dass ich mich nicht sehr gut mit der Syntax auskenne und hier am meisten Zeit spare.

Klar wurde aber auch: Ohne Grundwissen zu ML und Modellinterpretation stößt man beim Prompting schnell an Grenzen.

Data-Driven Personas und Data Storytelling

Im Marketing sind Personas ein etabliertes Mittel, um Zielgruppen greifbar zu machen und zum Beispiel die Kommunikationsmaßnahmen besser auf ihre Bedürfnisse zuzuschneiden. Ich ließ mir zunächst generische Personas basierend auf dem Datensatz erstellen – später auch spezifische Varianten pro Warengruppe, sogenannte Data-Driven Perso-



Abbildung 9: Infografik einer Persona (Quelle: Fabian Heidenstecker)

nas. Statt auf subjektiven Einschätzungen basieren diese auf realen Daten.

Nachdem ich grundsätzliche Informationen zum Aufbau einer guten Data Story erhalten hatte, experimentierte ich mit einigen Prompts zur Erzeugung der Personas. Dies waren für meinen Praxistest die umfangreichsten Prompts, daher dient der folgende Prompt als verkürztes Beispiel.

Prompt: „Erzeuge 5 Data-Driven Personas basierend auf dem Kaufverhalten von Kund:innen.“

Ergebnis: tabellarische Personas mit Attributen wie Alter, Einkommen, bevorzugtem Kanal und Ausgabenverhalten. Ergänzend generierte ChatGPT auch passende Handlungsempfehlungen für das Marketing.

Da ChatGPT auch Bilder erstellen kann, ging ich noch den letzten Schritt und erstellte eine Infografik (siehe Abbildung 9).



ge – mit teils überraschend guten Ergebnissen. Für technisch versierte Anwender bedeutet das vor allem: Geschwindigkeit, Flexibilität und neue Spielräume für explorative Analysen.

Doch die Risiken sind nicht zu unterschätzen: KI erfindet mitunter Informationen, rechnet nicht wirklich und ist abhängig von klarer Kommunikation. Gerade in sicherheitskritischen oder regulierten Umgebungen ist ein sorgfältiger Review unerlässlich. Auch der Datenschutz muss mitgedacht werden – besonders bei sensiblen Informationen.

Sensible Daten schützen

Auch wenn in meinem Beispieldatensatz keine Klarnamen oder direkte Identifikatoren enthalten sind, handelt es sich bei den Daten um sensible Informationen – schließlich geben Merkmale wie Einkommen, Alter oder Kaufverhalten Rückschlüsse auf reale Personen. Der verantwortungsvolle Umgang mit solchen Daten ist daher unerlässlich.

Bei der Nutzung generativer KI zur Analyse sollte daher immer sichergestellt sein, dass keine personenbezogenen Daten ungeschützt verarbeitet oder weitergegeben werden. Der Einsatz generativer KI – insbesondere cloudbasierter Lösungen – erfordert eine präzise Betrachtung hinsichtlich DSGVO, Speicherort, Zugriffskontrolle und Anonymisierung.

Tipp: Frühzeitig Datenschutzmaßnahmen wie Pseudonymisierung oder On-Premise-Verarbeitung einplanen.

Fehlerquellen im Blick behalten

Die GenAI operiert mit Wahrscheinlichkeiten. Von Mathematik besitzt sie nur ein begrenztes Verständnis. Insbesondere bei statistischen Analysen kommt sie daher an ihre Grenzen, was sie so direkt nicht unbedingt zugibt.

Wir dürfen nicht vergessen, dass die Sprachmodelle in erster Linie dazu da sind, flüssige und gut klingende Sprache zu generieren. Ob die Inhalte korrekt sind, ist für sie dabei zweitrangig, teilweise werden diese auch einfach frei erfunden. Das kennen wir dann unter den Begriff „Halluzinationen“.

Was bedeutet das zum Beispiel für unseren Python-Code?

Ein klassisches Beispiel ist die Berechnung von Durchschnittswerten. Die sind zuweilen gar nicht so intuitiv berechenbar, da sich der Durchschnitt immer auf etwas bezieht: Durchschnitt pro

Zwischenfazit: Die Ergebnisse sind nützlich – wenn auch mit Einschränkungen:

- Mehrere Infografiken gleichzeitig lassen sich nicht generieren.
- Stil und Inhalt der generierten Bilder sind inkonsistent.
- Textliche Beschreibung und Attributwerte passen nicht immer zusammen.

Der Einsatz von ChatGPT bietet in diesem Fall einen guten Startpunkt für die datengetriebene Segmentierung.

Grenzen und Verantwortung

ChatGPT verändert die Art, wie wir mit Daten interagieren. Statt sich durch komplexe BI-Tools zu klicken oder SQL-Abfragen zu schreiben, reicht oft eine einfache Fra-

Zeiteinheit, Durchschnitt über alle Kunden, Durchschnitt aller Zahlen und vieles mehr.

Drückt man sich nicht klar genug aus, dann kann hier eine falsche Annahme erfolgen, und ein syntaktisch richtiger Python-Code kann trotzdem ein ungenaues oder sogar falsches Ergebnis liefern.

Tipp: Nicht blind auf die KI vertrauen, sondern das eigene Zahlenverständnis und die eigene Einschätzung mit einsetzen! Im Zweifel würde ich dazu raten, den Code extern auszuführen und zum Beispiel Zwischenergebnisse der Berechnungen ausgeben zu lassen, um diese nachzuprüfen.

Fazit: KI kann viel – aber nicht alles

Generative KI ermöglicht heute bereits leistungsstarke, intuitive Datenanalysen – mit hoher Zugänglichkeit und erstaunlicher Effizienz. Doch: Sprachmodelle operieren mit Wahrscheinlichkeiten, nicht mit „Verständnis“. Fehlinterpretationen und Halluzinationen sind keine Seltenheit.

Gerade bei numerischen Werten ist Präzision gefragt. Unklare Prompts können zu korrektem, aber semantisch falschem Code führen. Deshalb gilt: Ergebnisse validieren, Code nachvollziehen und bei Bedarf lokal prüfen.

Trotz aller Faszination bleibt also ein realistischer Blick wichtig:

- „Rechnen“ heißt nicht „verstehen“ – Durchschnittswerte können falsch interpretiert werden, wenn der Prompt unklar ist.
- Code ist nicht immer korrekt – auch syntaktisch einwandfreier Code kann logisch falsch sein.
- Datenschutz ist Pflicht – gerade bei cloudbasierten KIs darf nie mit personenbezogenen Daten gearbeitet werden

Wo stehen wir aktuell?

Für technisch versierte Anwenderinnen und Anwender bietet ChatGPT ein mächtiges Werkzeug zur explorativen Analyse und Visualisierung von Daten – gerade in frühen Projektphasen. Die Fähigkeit, in natürlicher Sprache komplexe Auswertungen durchzuführen, spart enorm Zeit und senkt die Einstiegshürden.

Ausblick: Integration in Enterprise-Umgebungen

Die natürlichsprachliche Interaktion mit Daten wird zukünftig eine wichtige Rolle spielen – auch in Oracle-zentrierten Architekturen. Denkbare Einsatzszenarien:

- Generative KI als Add-on für Oracle Analytics Cloud
- Prompt-basierte Analyse in Kombination mit Autonomous Data Warehouse
- Ergänzung zu klassischen Tools wie APEX

Entscheidend ist: No Data – No AI. Daten müssen gut aufbereitet und verfügbar sein, um auch in einer explorativen Datenanalyse nutzbar zu sein.

Wichtig ist dabei immer: Die KI ersetzt keine Fachlichkeit – aber sie reduziert Hürden und beschleunigt Prozesse.

Mit zunehmender Reife der Werkzeuge wird die Grenze zwischen Analyse und Kommunikation weiter verschwimmen – und genau hier liegt das strategische Potenzial von GenAI.

Quellen

Prompts, Dataset und komplette Analyse:
https://github.com/Fabster79/redstack_eda

Über den Autor

Fabian Heidenstecker ist seit circa 20 Jahren in der IT unterwegs. Die letzten 15 Jahre in unterschiedlichsten Positionen im Consulting. Ursprünglich startete er im CRM-Umfeld und kam dort erstmals mit dem Thema BI und Analytics in Berührung. So wurden Daten seine Leidenschaft, welcher er in zahlreichen Projekten nachgehen konnte. In den letzten Jahren kamen noch die Themen Machine Learning und künstliche Intelligenz hinzu.

Heute ist er Senior Manager für Solutions bei Opitz Consulting und begleitet mit seinem Team Kunden auf dem Weg zur Data-Driven Company.



Fabian Heidenstecker
 fabian.heidenstecker@
 opitz-consulting.com