Interhyp

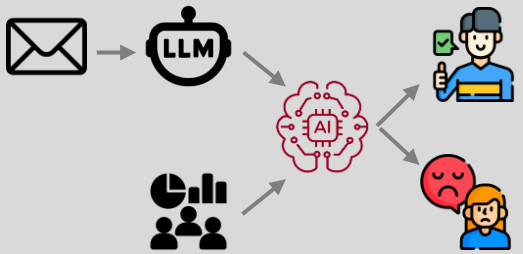
Case Study

Beschreibung des Use und Business Case

Durch die frühzeitige Erkennung unzufriedener Kunden können Gegenmaßnahmen eingeleitet und somit bestenfalls ein Nichtabschluss verhindert werden

Ausgangslage

Mittels **LLM-basierten (E-Mail) und Kunden-Features** (demographische Daten) soll ein **Klassifikationsmodell zur Vorhersage der Kundenzufriedenheit** mit der Interhyp erstellt werden.



Zielvariable

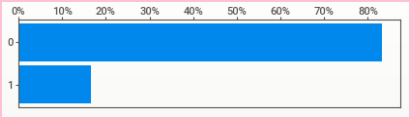
Da das vorrangige Ziel sein sollte, **nicht zufriedene Kunden möglichst frühzeitig** zu erkennen, wird der **NPS-Score so binärisiert**, dass Kunden mit einem **Score von 0 oder weniger die positive Klasse** bilden.

100	1,523	83.2%
0	204	11.1%
-100	102	5.6%
50	2	0.1%

0	1,525	83%
1	306	17%

Herausforderungen

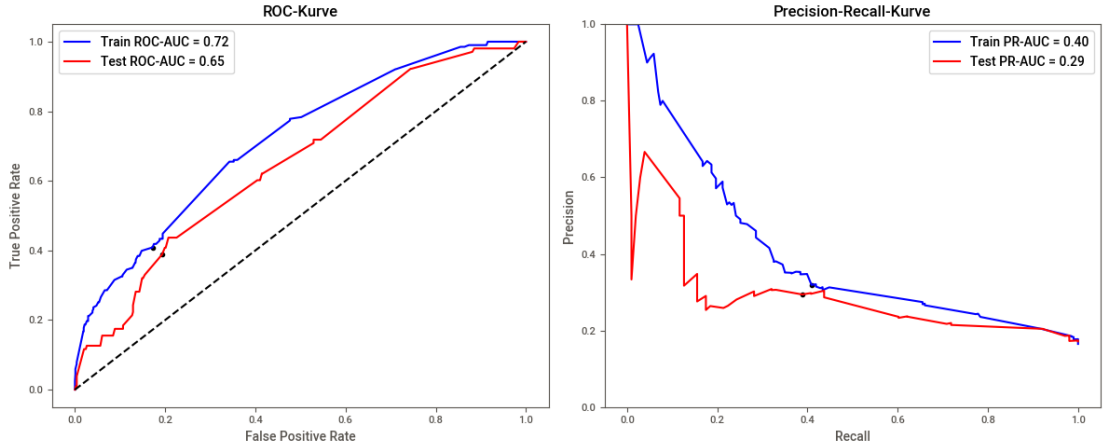
Die **Zielvariable weist eine recht hohe Imbalance** auf, welche (vor allem) in der Evaluation berücksichtigt werden muss.



Kunden können im Datensatz mehrfach vorkommen (doppelte app_ids). Da die Beobachtungen dann nicht unabhängig voneinander sind, muss darauf geachtet werden**, dass ein Kunde nur in Train oder Test vorkommt**.

Evaluation der Modellgüte

Im Vergleich zur Logistischen Regression als Baseline Model zeigt das LightGBM bessere Performance, auch wenn noch weiter Overfitting zu korrigieren ist

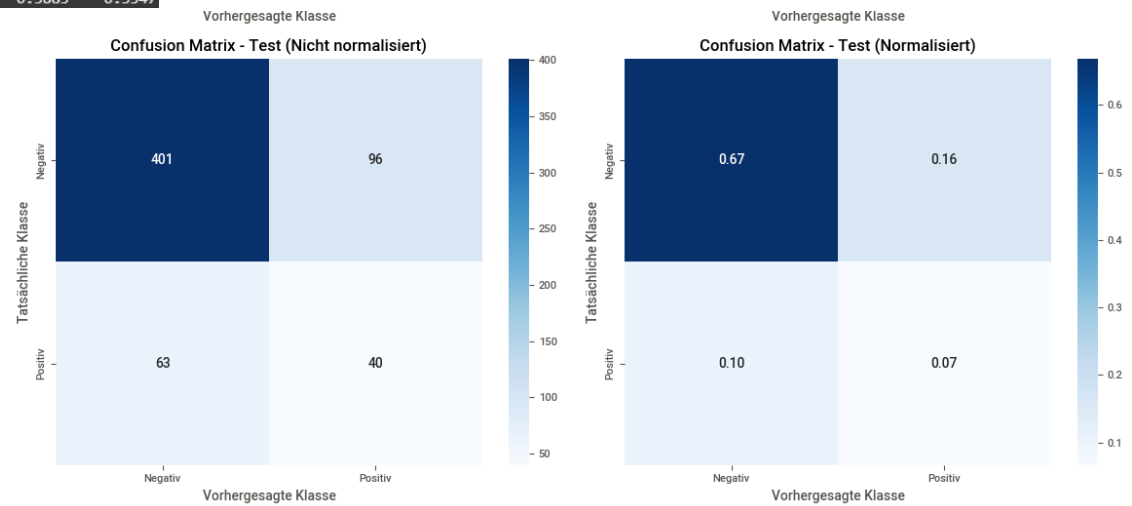


Um der **Imbalance des Datensatzes** gerecht zu werden, werden **ROC-AUC und PR-AUC als Metriken** herangezogen, um die Modellperformance zu messen. Es wird ersichtlich, dass eine **Vorhersage unzufriedener Kunden mittels der vorliegenden Daten durchaus passabel** funktioniert.

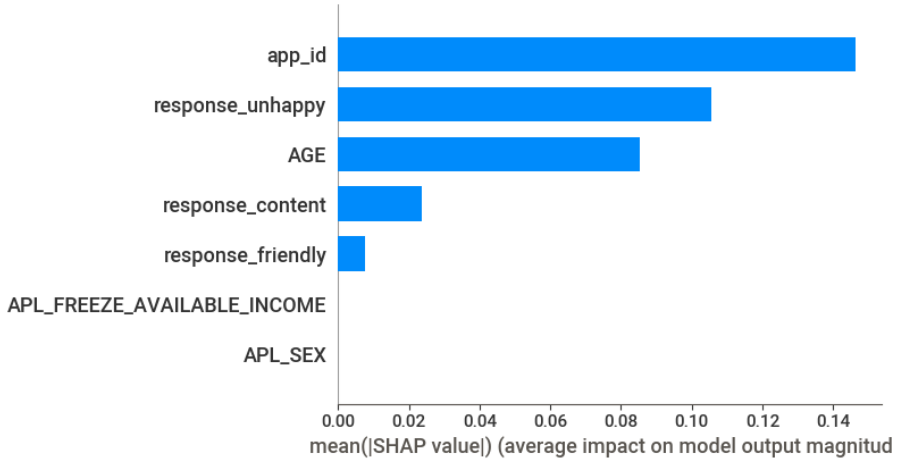
	Accuracy	Precision	Recall	F1-Score
Train	0.7587	0.3192	0.4089	0.3585
Test	0.7350	0.2941	0.3883	0.3347

Wenn man die **20% der Vorhersagen mit dem höchsten Score betrachtet** (Threshold bei 0.178), können im Test-Set von den **103 Positives auch 40 als solche erkannt (True Positives)** werden. Dabei werden allerdings auch **96 False Positives** erzeugt.

Bei einer zufälligen Vorhersage und der gleichen Menge an positiven Vorhersagen wären es nur ca. 23 True Positives gewesen (17% positive Samples bei 136 positiven Vorhersagen).

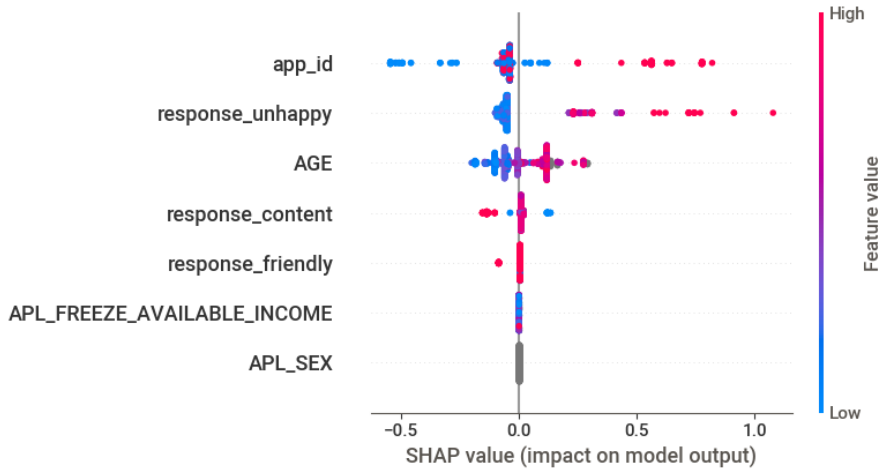


app_id und Alter sind neben dem LLM-basierten Feature response_unhappy für das trainierte LightGBM die wichtigsten Features

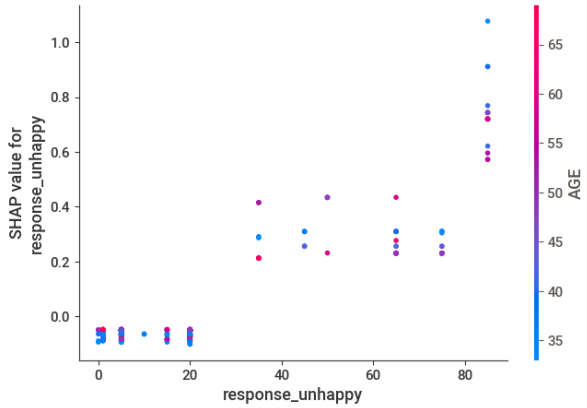


Die **app_id trägt im Mittel am meisten zur Modellvorhersage bei** (z.B. über Ausdruck einer gewissen Produktkategorie). Von den **LLM-basierten Features ist für das Modell response_unhappy die wichtigste** Kennzahl. Auch das **Alter hat allerdings durchaus Einfluss** darauf, ob ein Kunde zufrieden oder unzufrieden ist.

Höhere app_ids und Alter erzeugen oft höhere Vorhersagewerte, während kl**einere app_ids und niedrigere Alter die Vorhersagewerte** je nach Konstellation der übrigen Feature-Ausprägungen des Samples **meist verringern**.

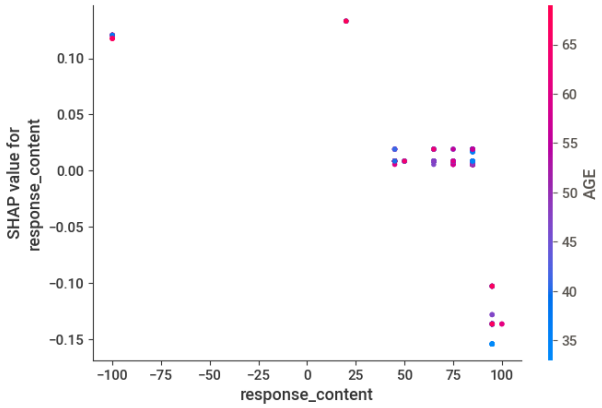


Die LLM-basierten Features wirken sich in erwarteter Weise auf die Modellvorhersage aus

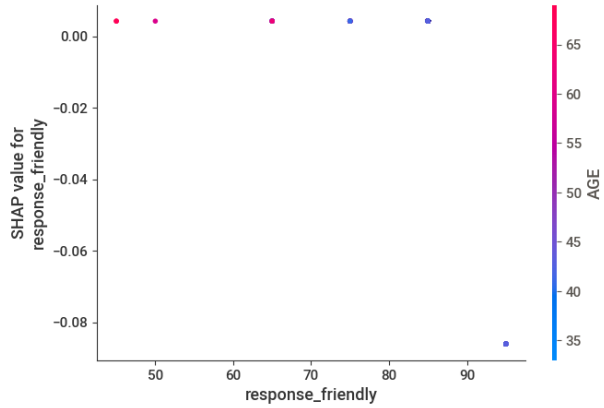


Je **unglücklicher das LLM den Text des Kunden einstuft, desto höher ist auch die Wahrscheinlichkeit**, dass der Kunde später tatsächlich unzufrieden mit der Interhyp war.

Der Impact auf die Vorhersage ist deutlich höher als bei den anderen LLM-Features.



Bei **extrem niedriger Zufriedenheit** innerhalb des Textes **erhöht dies eher die Wahrscheinlichkeit für einen schlechten NPS-Score**, während eine **extrem hoher response_content Wert eher mit einem hohen NPS-Score einhergeht**.



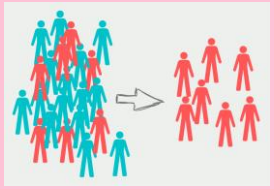
Da das Feature hinsichtlich Freundlichkeit für das Modell **nicht von großer Bedeutung** war, zeigen sich hier auch kaum Einblicke. **Eine sehr hohe Freundlichkeit innerhalb des Textes verringert jedoch die Wahrscheinlichkeit** laut dem Modell etwas, dass der Kunde hinterher unzufrieden mit der Interhyp war.

Das generierte Modell eignet sich unter Umständen nur bedingt für einen produktiven Rollout und bietet Potenzial zur Verbesserung

NPS als Target

Da der **NPS-Score das Target** des Datensatzes bildet, ist davon auszugehen, dass der **Datensatz nicht repräsentativ für den gesamten Kundenstamm ist**.

Der **NPS-Score wird für gewöhnlich von Kunden freiwillig erhoben**, sodass hier ein gewisser **Selbstselektionseffekt auftreten dürfte, welcher zu Bias** führt.



LLM-Features

Da für das Training **der zwischengeschalteten LLM-Modelle die Generierung von Labels erforderlich** gewesen sein muss, fließt eine gewisse **subjektive Einschätzung von Emotionen und somit auch die Fehler bei dieser** mit in die Modellausgabe (egal ob vortrainiert, selbst trainiert oder fine-getunet).

Diese Subjektivität (es wird eingeschätzt wie jemand anderes empfunden hat) **lernt das Modell nun mit, wenn es auf einem weitestgehend objektiven Target trainiert wird** (Kunde gibt sein tatsächliches Empfinden im NPS-Score an).

Das generierte Modell eignet sich unter Umständen nur bedingt für einen produktiven Rollout und bietet Potenzial zur Verbesserung

Abschluss als Target

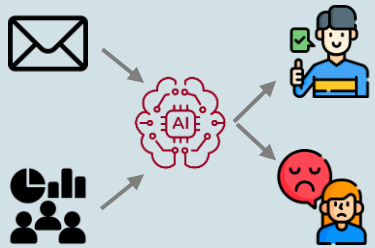
Wenn man direkt **den Abschluss als Target** nehmen würde, würde dies das **Problem der verzerrten Daten lösen**. Es wäre zwar dann immer noch **nicht zwangsläufig repräsentativ für alle Kunden, jedoch für alle Kunden, die E-Mails schreiben**, was für den Use Case ausreichend wäre. Somit sollte auch die verfügbare **Sample Size steigen**.

Da Zufriedenheit der Kunden (Weiterempfehlung vernachlässigt) bei der Interhyp vermutlich eine geringere Rolle spielt (Einmalkunden), sollte **das Target Abschluss auch die wichtigere KPI** abbilden.

E-Mail-Text direkt ins Modell

Wenn der **E-Mail-Text unmittelbar in ein mit dem Target verknüpftes LLM** fließen würde, wäre das Modell besser in der Lage, mit dem jeweiligen Target verknüpfte **Muster in den Formulierungen zu erkennen** und dafür relevante Features direkt aus dem Text zu extrahieren.

Eine **Anreicherung mit weiteren Kundendaten oder auch den LLM-Features wäre weiterhin möglich** (z.B. Concat vor Dense Layern).



Weitere Features

Weitere **Features, über den Kunden, welche eine Verbesserung des Profilings möglich machen und gut zugänglich** sein sollten. Diese wären z.B.:

- Wohnort
- Kontaktkanal
- Kontakthistorie
- Gewünschter Kreditrahmen

Mittels **LLM könnten weitere Features generiert werden, wie bspw. der Kontaktgrund** (Intenterkennung).

Potenzielle weitere Use Cases im E-Mail-Bereich reichen von der Nutzung von Insights bis hin zu einer zeitlich kausalen Optimierung der Beantwortung

Insightgenerierung: Clusterbildung innerhalb der E-Mails, um auf Management-Ebene zu wissen, welche Anfrage überhaupt eingehen bzw., ob sich diese über den Zeitverlauf ändern.
→ Erkennung von Schulungspotenzial oder Priorisierung von Clustern im Dokumenteneingang

Techn. Umsetzung: z.B. Embedding (SentenceTransfomer) → Dimensionsreduktion (UMAP) → Clustering (HDBSCAN) → Beschreibung des Clusters (LLM)

Halbautomatisierte E-Mail-Beantwortung mittels RAG: Für eingehende E-Mails könnten Antworten vorproduziert werden, welche im Nachgang nur noch angepasst werden müssten. Hierbei kann RAG unterstützen, um Informationen über Produkte (P) oder Arbeitsanweisungen (AA) zu liefern.
→ Effizienzsteigerung in der Kundenbetreuung

Techn. Umsetzung: z.B. (Embedding E-Mail ← Matching → Embedding P & AA) → Bereitstellung relevanter Infos aus P & AA → Antwortgenerierung (LLM)

Intentbedingtes E-Mail-Routing: Für eingehende E-Mails könnte der Intent der Anfrage automatisch erkannt werden und anschließend direkt den dafür zuständigen Spezialisten zugesteuert werden.
→ Effizienzsteigerung in der Kundenbetreuung und ggf. Erhöhung der Kundenzufriedenheit

Techn. Umsetzung: z.B. Insightgenerierung → Zusammenfassen von Clustern nach Themengebieten (manuell) → Training eines Modells mit Clustern als Target (LLM)

Kausal priorisierte E-Mail-Bearbeitung: Verschiedene Kundentypen erwarten unter Umständen eine Beantwortung einer Mail in einem unterschiedlichen Zeitrahmen (Heterogenität). Vor allem bei Rückstandssituationen in der E-Mail-Bearbeitung könnte eine zeitlich kausale Priorisierung der E-Mail-Beantwortung stattfinden.
→ Erhöhung der Kundenzufriedenheit

Techn. Umsetzung: z.B. (zeitlich) randomisiertes Experiment → Training eines kausalen Modells (CausalForest) → Schätzung von Conditional Average Treatment Effects (CATE) → Nutzung der CATE-Schätzungen für die heterogene Optimierung