INTERNATIONALE HOCHSCHULE FERNSTUDIUM

Fallstudie - DLMDWME01

University of Applied Science – online

Studiengang: Data Science Master

DLMDWME01– Model Engineering

Clara Simon

Matrikelnummer: 321150405

23.10.2025

Neubulach

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Formelverzeichnis

Abkürzungsverzeichnis

Explorative Datenanalyse ..... EDA
Payment Service Provider ..... SPS
Team Data Science Process ..... TDSP

1 Einleitung

Im Rahmen meiner Tätigkeit als Data Scientist bei einem der weltweit größten Einzelhandelsunternehmen wurde ich mit der Analyse und Optimierung von Online-Zahlungsprozessen betraut. Diese Hausarbeit dokumentiert die Entwicklung eines Machine-Learning-basierten Empfehlungssystems, das darauf abzielt, die Effizienz von Transaktionen durch eine intelligente Auswahl des Zahlungsdienstleisters zu steigern.

1.1 Motivation und Zielsetzung

Das Unternehmen steht vor der Herausforderung einer ineffizienten Zahlungsabwicklung, insbesondere bei Online-Kreditkartenzahlungen, die im letzten Jahr eine hohe Ausfallrate aufwiesen. Eine Analyse zeigt, dass 79,7 % aller Zahlungsversuche fehlschlagen. Diese hohe Fehlerrate führt zu direkten finanziellen Verlusten durch anfallende Gebühren an die vier Zahlungsdienstleister und verursacht zusätzlich erhebliche strategische Schäden. Letzteres resultiert aus einer negativen Kundenerfahrung, die Kaufabbrüche, Umsatzverluste und eine nachhaltige Schädigung der Kundenbindung zur Folge hat. Ziel dieser Arbeit ist daher die Entwicklung eines intelligenten, automatisierten PSP-Routing-Systems. Dieses Prognosemodell soll mittels Machine-Learning-Methoden in Echtzeit den optimalen Zahlungsdienstleister für jeden Versuch vorschlagen. Die Optimierung verfolgt dabei die Maximierung der Transaktionserfolgsrate (als Maß für die Kundenzufriedenheit) und die Minimierung der Transaktionsgebühren.

1.2 Aufbau der Arbeit

Die vorliegende Arbeit ist angelehnt an den Team Data Science Process strukturiert und gliedert sich in die entsprechenden Phasen dieses Vorgehensmodells (Microsoft, n.d.). Nach der Einleitung (Kapitel 1), welche die Motivation und Zielsetzung des Projekts darlegt, wird in Kapitel 2 das Geschäftsverständnis vertieft. Kapitel 3 befasst sich mit dem verwendeten Ansatz zur Versionierung des Projekts. Im zentralen Kapitel 4, Datenerfassung und Verständnis, wird die Datenqualität analysiert und die Datenvorverarbeitung detailliert beschrieben, einschließlich der Merkmalgenerierung, Kodierung und Skalierung numerischer Features. Den Abschluss dieses Abschnitts bildet die Explorative Datenanalyse, die gesondert für die Zielvariablen Success und Fee durchgeführt wird. Kapitel 5, die Modellierung, beginnt mit der Feature Selektion. Daraufhin folgt das Modell Training und die Evaluation für die beiden separat entwickelten Modelle: das Klassifikationsmodell und das Regressionsmodell. Abschließend wird die Funktionsweise des Kombinierten Modells dargelegt. Kapitel 6 beschreibt die Bereitstellung (Deployment) des entwickelten Systems und

Kapitel 7 die Evaluation dessen im Praxiseinsatz. Die Arbeit schließt mit dem Fazit (Kapitel 8), welches die Ergebnisse zusammenfasst und einen Ausblick auf zukünftige Entwicklungen gibt.

2 Geschäftsverständnis

Als Grundlage dieses Projektes, ist die Definition einer Transaktion und die, der dazugehörigen Versuchen essentiell. Ein einzelner Kaufvorgang (Transaktion) kann mehrere Zahlungsanläufe (Versuche) umfassen, von denen jeder einzelne, ob erfolgreich oder nicht, Gebühren verursacht.

PSP	Gebühr erfolgreicher Versuch	Gebühr fehlgeschlagener Versuch
Moneycard	5€	2€
Goldcard	10€	5€
UK Card	3€	1€
Simplecard	1€	0,5€

Tabelle 1 Kosten der Versuche je PSP (aufgeteilt in Success und Fail)

Die Gebührenstruktur (Tabelle 1) variiert dabei erheblich zwischen den Anbietern: Goldcard ist mit 10 € pro erfolgreichen Versuch am teuersten, während Simplecard mit nur 1 € die kostengünstigste Option darstellt.

Das primäre Ziel des Unternehmens ist die Stärkung der Kundenzufriedenheit (Priorität 1), gefolgt von der Senkung der Kosten (Priorität 2). Das Kernproblem stellt somit eine Optimierungsaufgabe dar, bei der die Wahrscheinlichkeit eines erfolgreichen Transaktionsabschlusses gegen die kumulierten Kosten der notwendigen Versuche abgewogen werden muss. Um dieses Problem zu lösen, wurden zwei separate Modelle entwickelt: ein Klassifikationsmodell zur Vorhersage der Erfolgswahrscheinlichkeit pro Zahlungsdienstleister (PSP) und ein Regressionsmodell zur Schätzung der voraussichtlichen Kosten pro PSP.

$$(\text{Klassifikationswahrscheinlichkeit} \times 0{,}7) - (\text{Normalisierte Kosten} \times 0{,}3).$$

Formel 1 Finale Modellgewichtung

Diese Ergebnisse werden, wie in Formel 1 dargestellt, kombiniert, um die finale Empfehlung zu generieren. Basierend auf den festgelegten Prioritäten wird dabei die Erfolgswahrscheinlichkeit mit 70 % und die Kosten mit 30 % gewichtet.

3 Dateistruktur und Versionierung

Für das Projekt wurde eine Aufteilung der Dateien verfolgt. Der Ordner Data beherbergt sämtliche Excel- und PKL-Datensätze. Die darauf aufbauende Vorverarbeitung erfolgt im Ordner data_preperation. Für die Feature Generierung und Selektierung ist der Ordner Feature_Engineering zuständig. Der Ordner data_analysis beinhaltet die Explorative Datenanalyse und den Unterordner plots mit allen generierten Darstellungen. Der Ordner

model dient der Entwicklung des Regressionsmodells, des Klassifikationsmodells und des finalen Modells. Er enthält die PKL-Dateien der finalen und der weiter betrachteten Modelle sowie einen Unterordner plots für die entsprechenden Visualisierungen.

Für das Training der Modelle wird die Open-Source-Plattform MLFlow genutzt (MLFlow, n.d.). MLflow ermöglichte die lückenlose Protokollierung aller durchgeführten Modelltrainings, inklusive der verwendeten Hyperparameter, Code-Versionen und resultierenden Modelle. Zudem wurde das gesamte Projekt in einem Git-Repository versioniert. Um das Repository schlank zu halten und einen zu großen Umfang zu vermeiden, werden alle MLflow-Ordner und Modellordner in der .gitignore-Datei ausgeschlossen.

4 Datenerfassung und Verständnis

Die Grundlage für das zu entwickelnde Prognosemodell bildet der strukturierte Datensatz PSP_Jan_Feb_2019.xlsx, welcher Kreditkartentransaktionen aus den DACH-Ländern für Januar und Februar 2019 enthält.

4.1 Datenqualität

Die initiale Datenvorbereitung orientiert sich an den ersten Schritten des CRISP-DM-Prozesses und umfasst sowohl die Untersuchung der Datenqualität als auch eine erste deskriptive Analyse (IBM, 2021).

4.1.1 Datenbereinigung und -struktur

Zunächst wurde der Rohdatensatz in einen Pandas DataFrame geladen, um die Datenstruktur zu prüfen und notwendige Bereinigungsschritte durchzuführen. Der initiale Umfang des Datensatzes betrug 50.410 Zeilen und 8 Spalten. Die Spalten umfassten 'Unnamed: 0', 'tmsp' (Zeitstempel), 'country', 'amount' (Überweisungsbetrag), 'success' (Zielvariable), 'PSP' (Payment Service Provider), '3D_secured' und 'card'. Im Rahmen der Datenqualitätsprüfung wurden 81 Duplikate in den relevanten Spalten ('tmsp' bis 'card') identifiziert und entfernt. Dies führte zu einem finalen Umfang von 50.329 Zeilen. Eine Überprüfung auf fehlende Werte (NaN) ergab, dass in keiner der Spalten Lücken vorhanden waren, was die Datenqualität in dieser Hinsicht als sehr hoch ausweist.

4.1.2 Deskriptive Analyse der Transaktionsdaten

Die deskriptive Analyse lieferte erste Einblicke in die Verteilung der numerischen und kategorialen Variablen. Die statistische Auswertung der Versuchsbeträge ergab folgende Kennzahlen: Der Mittelwert lag bei 202,38 € und der Median bei 201,00 €. Die nahezu identischen Werte von Mittelwert und Median deuten auf eine nahezu symmetrische

Verteilung der Beträge hin. Die Standardabweichung betrug 96,26 €, was auf eine erhebliche Streuung der Versuchsbeträge hindeutet. Mit einem Minimum von 6,00 € und einem Maximum von 630,00 € (25. Perzentil: 133,00 €, 75. Perzentil: 269,00 €) sind keine offensichtlichen, extremen Ausreißer festzustellen; der Maximalbetrag liegt im plausiblen Rahmen für Kreditkartentransaktionen dieser Art.

4.1.3 Analyse der kategorialen Spalten

Im Rahmen der Datenuntersuchung wurde die Analyse der kategorialen Spalten durchgeführt, um die Plausibilität der Ausprägungen und deren Verteilung zu bestätigen. Die Transaktionen stammen aus den drei DACH-Ländern, wobei Deutschland mit einem Anteil von knapp 60 Prozent (59,97%) die klare Dominanz aufweist. Die Schweiz und Österreich folgen mit ähnlichen Anteilen von 20,51% bzw. 19,51%. Hinsichtlich der verwendeten Payment Service Provider existieren vier verschiedene Anbieter, unter denen UK Card mit 52,43% die Hälfte aller Versuche abwickelt. Die weiteren PSPs verteilen sich auf Simplecard (24,72%), Moneycard (16,48%) und Goldcard (6,37%). Bei der Kartenart dominiert die MasterCard mit über der Hälfte der Fälle (57,52%), während Visa (23,10%) und Diners (19,38%) seltener verwendet werden. Des Weiteren wurde das 3D Secure-Verfahren in der überwiegenden Mehrheit der Versuche (76,17%) nicht angewandt, im Gegensatz zu 23,83% der Fälle. Eine zentrale Erkenntnis ergab die Analyse der Zielvariablen 'success': Die Daten weisen eine deutliche Ungleichverteilung der Klassen auf. Lediglich 20,32% der Versuche waren erfolgreich, während fast 80 Prozent (79,68%) fehlschlugen.

4.1.4 Zeitreihenanalyse ('tmsp')

Bezüglich der zeitlichen Verteilung auf Jahre und Monate stammen alle Einträge erwartungsgemäß aus dem Jahr 2019 und verteilen sich relativ gleichmäßig auf Januar (52,2%) und Februar (47,8%). Die Betrachtung der Wochentage zeigt, dass die Aktivität an Werktagen höher ist und ihren Höhepunkt am Dienstag (17,2%) erreicht. Zum Wochenende hin nimmt Diese kontinuierlich ab und erreicht ihren Tiefpunkt am Sonntag (10,6%). Über den Tagesverlauf betrachtet, sind die Versuche sehr gleichmäßig verteilt, mit einem konstanten Anteil von etwa vier Prozent pro Stunde.

4.2 Datenvorverarbeitung

Die Datenvorverarbeitung ist ein entscheidender Schritt vor der eigentlichen Modellentwicklung und umfasst mehrere vorbereitende Maßnahmen. Dazu gehört die Datenanreicherung, also das Hinzufügen oder Zusammenführen des Ausgangsdatensatzes mit weiteren relevanten Informationen, um den Informationsgehalt zu erhöhen. Weiterhin

müssen kategoriale Merkmale mithilfe des Feature Encoding in ein numerisches Format umgewandelt werden, damit maschinelle Lernalgorithmen sie verarbeiten können. Schließlich dienen die Skalierung und Normalisierung dazu, die Wertebereiche der numerischen Merkmale anzugleichen, wodurch die Daten für die Algorithmen vergleichbarer gemacht und die Modellleistung optimiert wird.

4.2.1 Merkmal Generierung

Transaktions-ID

Ein zentraler und geschäftsrelevanter Schritt der Datenvorverarbeitung war die Identifikation und Gruppierung von Mehrfachversuchen für denselben Einkaufsvorgang. Hierfür wurde vorgegeben, dass Transaktionen, die innerhalb derselben Minute, aus demselben Land und mit demselben Betrag stattfinden, zum selben Kaufversuch gehören. Um dies umzusetzen, wurde zunächst der Zeitstempel ('tmsp') auf die Minute gerundet. Anschließend wurde über die Kombination mit dem Land ('country') und dem Betrag ('amount') eine eindeutige transactionId zugewiesen.

Fee

Als nächstes wurde ein numerisches Feature zur Abbildung der Gebühren erstellt. Hierzu wurde die Gebühr pro Versuch hinzugefügt, basierend auf dem jeweiligen PSP und dem Erfolgsstatus (Tabelle 1).

Generierung wichtiger Geschäfts- und Zeit-Features

Darüber hinaus wurden umfangreiche zeitbasierte Features aus dem ursprünglichen Zeitstempel extrahiert, um tägliche, wöchentliche und saisonale Muster für das Modell zu analysieren. Konkret wurden der Wochentag (weekday und weekday_num), die Kalenderwoche (calendar_week), der Monat (month) und die Stunde (hour) abgeleitet. Zusätzlich erfolgte eine kategorische Einteilung der Stunde in die Tageszeiten Nacht, Morgen, Nachmittag und Abend (daytime). Außerdem wurde ein boolesches Feature (feiertag_DE) zur Markierung bundesweiter deutscher Feiertage ergänzt.

Amount

Abschließend wurde der kontinuierliche Transaktionsbetrag amount diskretisiert. Mithilfe der Quartile wurde die Spalte in vier Kategorien (amount_cat) von Q1_lowest bis Q4_highest unterteilt.

4.2.2 Kodierung

Da viele Machine-Learning-Modelle nur mit numerischen Eingaben arbeiten können, ist die Umwandlung in dieses Format essenziell. Die Wahl der geeigneten Kodierungsstrategie für die kategorialen Merkmale ist dabei entscheidend, um die tatsächlichen Beziehungen im Datensatz korrekt abzubilden und künstliche Rangfolgen zu vermeiden (Thakur, 2025). Für nominale Merkmale ohne natürliche Ordnung, wie das Land (country), der PSP (PSP), die Kartenart (card), der Monat (month) und der Wochentag (weekday), wurde das One-Hot Encoding (mit Präfix: oh__) angewendet. Hierbei wird für jede Kategorie des ursprünglichen Merkmals eine neue binäre Spalte (entweder 0 oder 1) erstellt, wodurch eine fälschliche numerische Gewichtung verhindert wird. Die einfachste Binäre Kodierung (mit Präfix: bin__) wurde für Merkmale verwendet, die von Natur aus nur zwei Ausprägungen besitzen. Dazu zählen die Zielvariable (success), die Verwendung von 3D Secure (3D_secured) und die Markierung eines Feiertags (feiertag_DE). Im Gegensatz dazu wurde für Merkmale mit einer klaren, logischen Reihenfolge die ordinale Kodierung (mit Präfix: ord__) gewählt. Dies betrifft die diskretisierte Transaktionshöhe (amount_cat), die nach Quartilen geordnet ist, sowie die logische Abfolge der Tageszeiten (daytime) und die Kalenderwoche (calendar_week). Abschließend kam die zyklische Kodierung (hour_sin) zum Einsatz, um den kreisförmigen Charakter der Stunde abzubilden. Da die Stunde 23 und die Stunde 0 (Mitternacht) zeitlich direkt benachbart sind, wurde die Variable mithilfe einer Sinus-Transformation kodiert (Valença, August). Die unveränderten Spalten wurden mit dem Präfix remainder__ ausgewiesen.

4.2.3 Normalisierung numerischer Features

Als letzter Schritt der Datenvorverarbeitung wurden die kontinuierlichen, numerischen Spalten remainder__amount und remainder__fee einer Skalierung unterzogen. Die Verteilung dieser Features wurde vorab analysiert, um die Wahl des geeigneten Skalierungsverfahrens zu begründen und potenzielle Ausreißer zu identifizieren (Jaiswal, 2025).

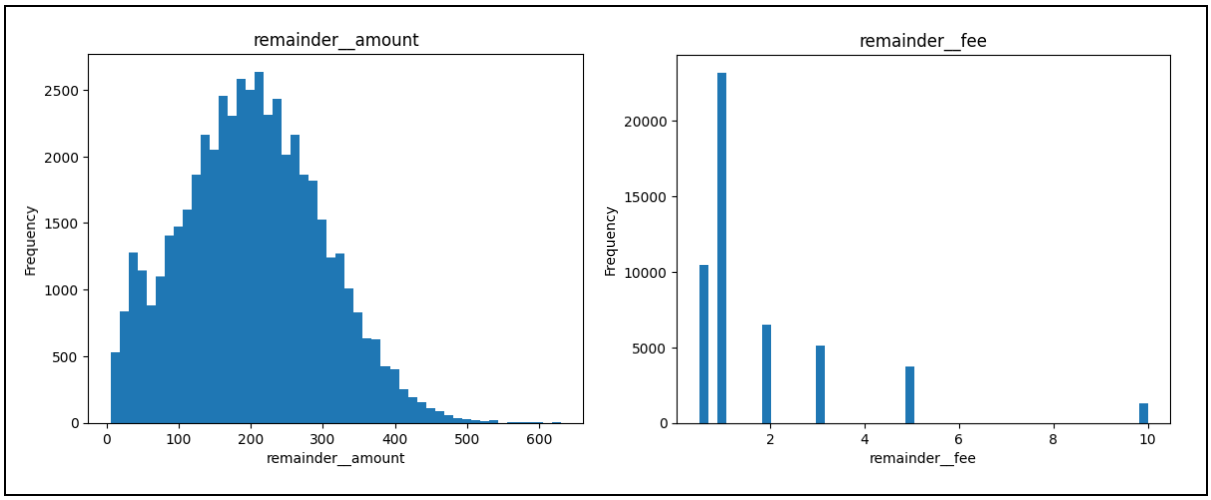

Abbildung 1 Verteilung Amount und Fee

Die Verteilung der Transaktionsbeträge, dargestellt im Histogramm von remainder__amount ist nahezu symmetrisch, weist jedoch einen leichten Rechtsschweif auf. Die Gebührenverteilung, ersichtlich im Histogramm von remainder__fee ist hingegen stark diskret und rechtsschief, da die Gebühren nur wenige spezifische Werte annehmen.

Um zu verhindern, dass die unterschiedlichen Wertebereiche dieser Features die Modellbildung dominieren und somit die Performance des Machine-Learning-Algorithmus verzerren, wurde die Min-Max-Skalierung (MinMaxScaler) gewählt (Jaiswal, 2025). Dieses Verfahren transformiert die Werte linear auf einen einheitlichen Bereich zwischen 0 und 1. Die skalierten Ergebnisse werden in den neuen Spalten remainder__amount_scaled und remainder__fee_scaled gespeichert.

Das finale Feature-Set, das nun eine optimierte Mischung aus binären, ordinalen, One-Hot-kodierten und skalierten numerischen Merkmalen, sowie den unveränderten Features umfasst, wurde abschließend in der Datei encoded_scaled_df.pkl gespeichert und ist damit bereit für die eigentliche Modellentwicklung.

4.3 Explorative Datenanalyse

Im Anschluss an die Datenvorverarbeitung dient die EDA dazu, die Daten zu untersuchen, ihre Struktur zu verstehen und erste Muster sowie Anomalien zu identifizieren. Der Fokus liegt dabei auf der detaillierten Analyse der Verteilung und der Beziehungen der Features, insbesondere in Bezug auf die beiden Zielvariablen success und Fee.

4.3.1 EDA für die Zielvariable success

Die erste explorative Datenanalyse widmet sich der Untersuchung von Zahlungsdaten mit dem Ziel, Muster und Einflussfaktoren zu identifizieren, die den Erfolg oder Misserfolg von Transaktionen, repräsentiert durch die Zielvariable success, bestimmen.

Gesamtbetrachtung der Transaktionen und Versuche

Eine initiale Analyse auf der Ebene der einzelnen Abläufe offenbart eine sehr hohe Fehlerrate: 79,7 % aller dokumentierten Zahlungsversuche schlugen fehl, während lediglich 20,3 % erfolgreich abgewickelt wurden. Wird hingegen die Gesamtheit der Transaktionen betrachtet, so ergibt sich ein etwas günstigeres, jedoch weiterhin klares Bild. Von insgesamt 37.825 Transaktionen wurden 10.227 (27,04 %) erfolgreich abgeschlossen – das heißt, in mindestens einem Anlauf innerhalb der Transaktion gelang die erfolgreiche Abwicklung. Die überwiegende Mehrheit von 27.598 Transaktionen (72,96 %) scheiterte vollständig, da jeder einzelne Durchgang fehlschlug. Dabei sind bis zu sechs aufeinanderfolgende Ausführungen innerhalb einer Transaktion zu beobachten.

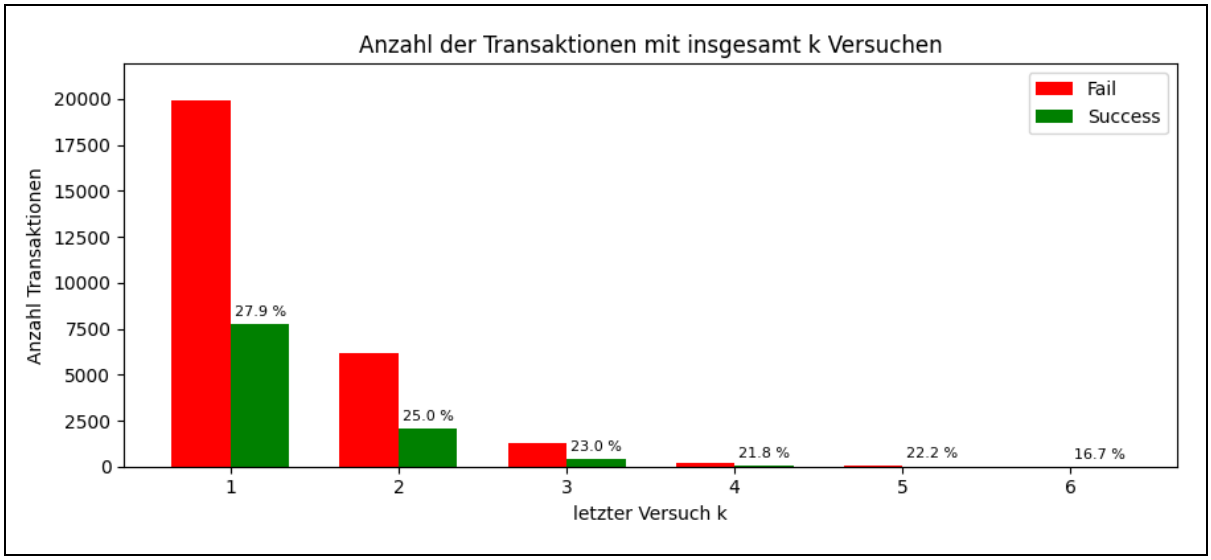

Abbildung 2 Anzahl der Transaktionen mit insgesamt k Versuchen (aufgeteilt in Success und Fail)

Die Verteilung der Transaktionen über die Anzahl der Abwicklungsversuche, wie in Abbildung 2 dargestellt, verdeutlicht die Dynamik des Prozesses. Der Großteil der Transaktionen wird bereits nach dem ersten Anlauf abgeschlossen. Mit jedem weiteren Ablauf sinkt das Volumen der Transaktionen rapide. Bemerkenswert ist, dass die prozentuale Erfolgschance mit zunehmender Zahl der Wiederholungsversuche tendenziell abnimmt. Eine Transaktion, die mehrere Durchläufe benötigt, hat demnach nicht nur bereits mehrfach keinen Erfolg erzielt, sondern auch eine schrittweise geringere Wahrscheinlichkeit für einen erfolgreichen nächsten Versuch. Der Prozess ist auf maximal sechs Ausführungen begrenzt.

Analyse der Transaktionsflüsse

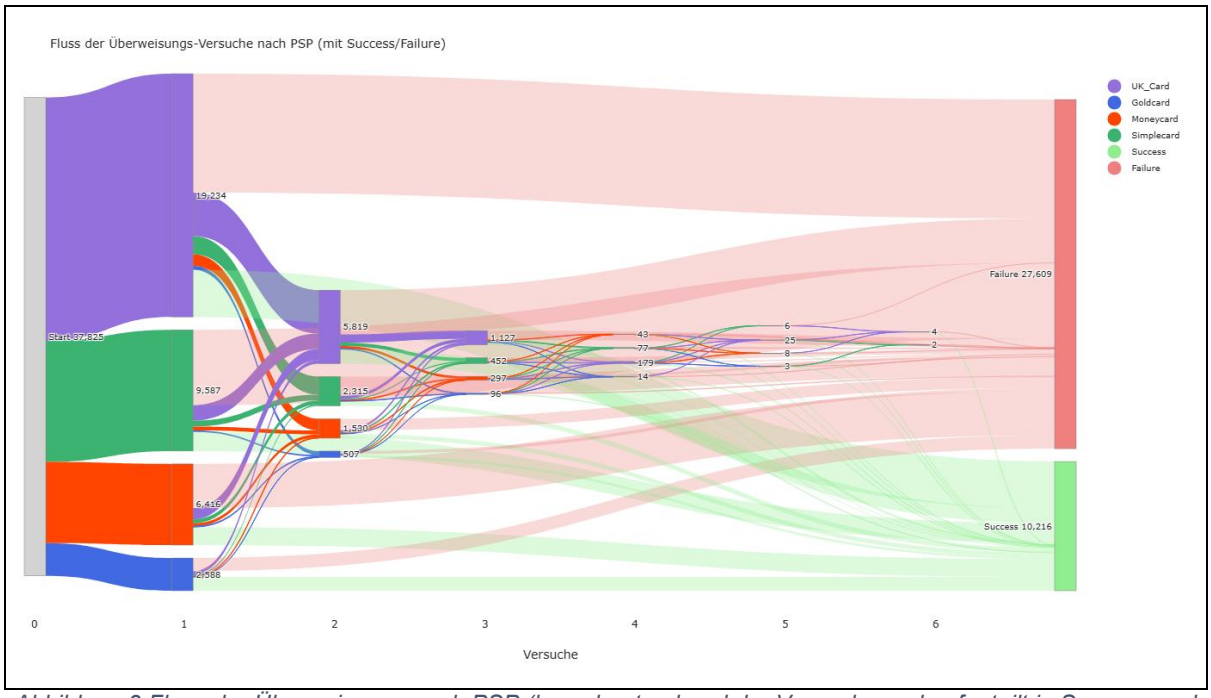

Abbildung 3 Fluss der Überweisungs nach PSP (berechnet anhand der Versuche und aufgeteilt in Success und Fail)

Die Visualisierung der Transaktionsflüsse mittels eines Sankey-Diagramms erlaubt tiefere Einblicke in die Verteilung und das Nutzerverhalten über mehrere Versuche hinweg. Das Diagramm 3 zeigt, dass die Zahlungsdienstleister UK Card und Simplecard die Erstversuche mit 19.234 bzw. 9.587 Transaktionen klar dominieren. Gleichzeitig sind es diese Anbieter, die auch die größten Ströme direkter und endgültiger Fehlschläge generieren, was durch die breiten, hellroten Bänder im Diagramm verdeutlicht wird. Goldcard hingegen, obgleich mit dem geringsten initialen Volumen, weist einen proportional höheren Erfolgsanteil auf. Nach dem ersten Versuch bricht das Volumen der weitergeführten Transaktionen --- ein.

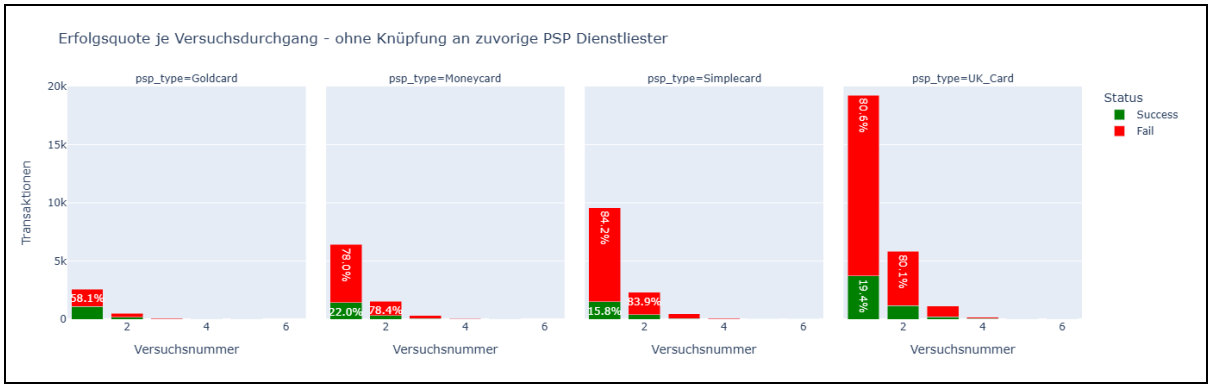

Abbildung 4 Erfolgsquote je PSP je Durchgang (berechnet anhand der Versuche und aufgeteilt in Success und Fail)

Eine quantitative Untermauerung dieser Beobachtung liefert die Abbildung 4, welche die klare Trennung der Zahlungsdienstleister in „Qualitäts-“ und „Quantitäts-Anbieter“ bestätigt. Hierbei kristallisiert sich Goldcard als der Anbieter mit der höchsten Effizienz heraus. Obwohl dieser

das geringste Transaktionsvolumen im ersten Versuch abwickelt, erzielt er mit einer Erfolgsquote von 58,1 % die mit Abstand beste Leistung. Demgegenüber stehen die volumenstarken Anbieter UK Card und Simplecard, die gleichzeitig die niedrigsten Erfolgsraten aufweisen. UK Card verzeichnet bei sehr hohem Aufkommen eine Erfolgsquote von lediglich 19,4 %, während Simplecard mit 15,8 % die geringste Effizienz im Vergleichsfeld zeigt. Moneycard positioniert sich mit einer Erfolgsrate von 22,0 % im mittleren Bereich zwischen diesen beiden Extremen.

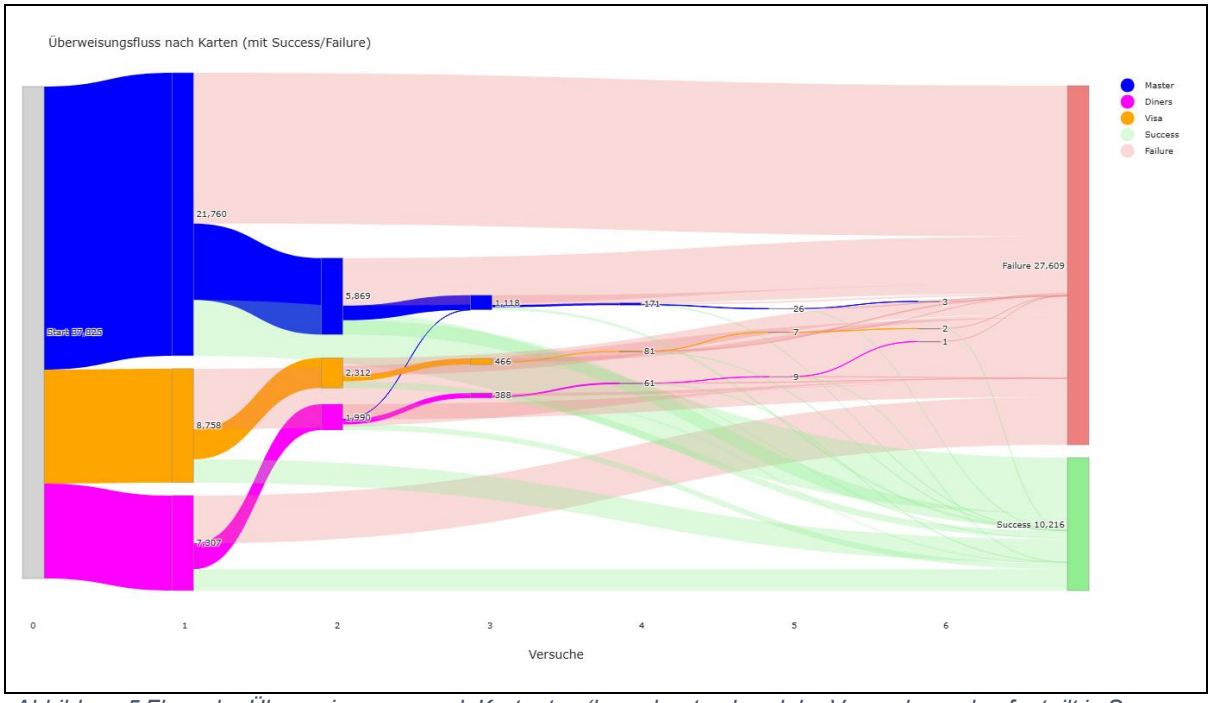

Abbildung 5 Fluss der Überweisungen nach Kartentyp (berechnet anhand der Versuche und aufgeteilt in Success und Fail)

Die Analyse nach Kartenart (Abbildung 5) offenbart eine überwältigende Dominanz der Master-Karte (blau), die als das klare "Arbeitspferd" des Systems fungiert. Mit 21.760 Transaktionen wickelt sie den Löwenanteil aller Erstversuche ab und bleibt auch in allen nachfolgenden Versuchen der volumenstärkste Kanal. An zweiter Stelle folgt die Visa-Karte (orange) mit 8.758 Erstversuchen, während die Diners-Karte (magenta) mit 5.587 Versuchen das geringste initiale Volumen aufweist. Die Kartentypen bleiben für jede Transaktion beibehalten. Ein sehr schmaler Fluss zeigt jedoch, dass in einem Fall der Kartentyp innerhalb einer Transaktion wechselt. Bei der Erstellung der Transaktions-IDs gab es voraussichtlich eine falsche Zuordnung, die zu dieser Anomalie führte.

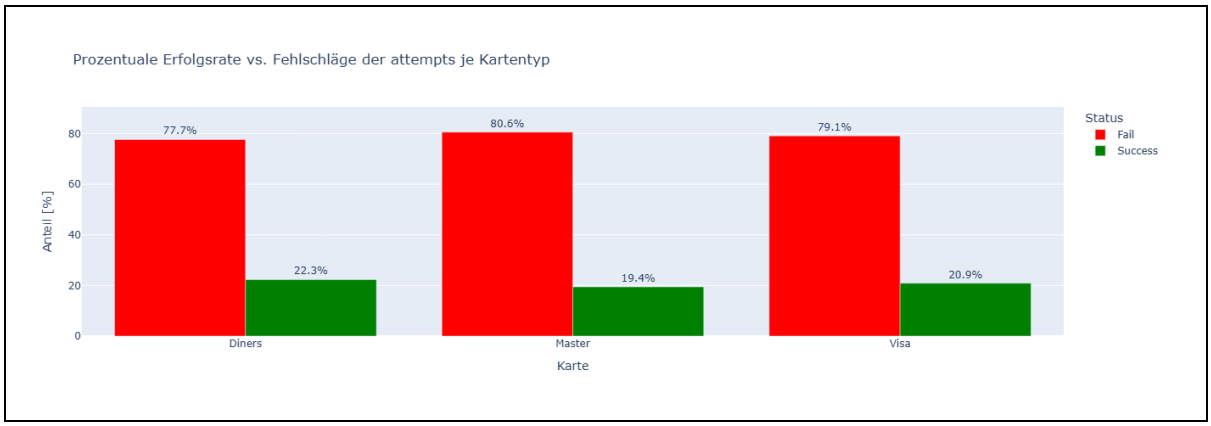

Abbildung 6 Versuche je Kartentyp (berechnet anhand der Versuche und aufgeteilt in Success und Fail)

Eine quantitative Ergänzung liefert die prozentuale Aufschlüsselung der Erfolgs- und Fehlerraten je Kartentyp (Abbildung 6). Obwohl es geringfügige Unterschiede gibt, bewegen sich alle Kartenarten auf einem sehr hohen Fehlerniveau. Die Master-Karte weist mit 80,6 % die höchste Fehlerrate und mit 19,4 % die niedrigste Erfolgsquote auf. Die Diners-Karte schneidet mit einer Erfolgsrate von 22,3 % marginal besser ab, dicht gefolgt von der Visa-Karte mit 20,9 %. Da die Erfolgswahrscheinlichkeit bei allen Kartentypen durchweg niedrig ist, lässt sich schlussfolgern, dass die Kartenart allein kein starker Prädiktor ist, um zwischen Erfolg und Misserfolg einer Transaktion zu unterscheiden.

Zeitliche Muster und Leistungscharakteristika

Die Untersuchung zeitlicher Merkmale war auf kurzfristige Muster beschränkt, da die Daten lediglich die Monate Januar und Februar 2019 umfassen und somit keine Analyse langfristiger saisonaler Trends ermöglichen. Dennoch lassen sich wöchentliche und tageszeitliche Muster erkennen, die auf spezifische Leistungscharakteristika der verschiedenen Kartenarten und Zahlungsdienstleister hindeuten.

Analyse nach Kartenart

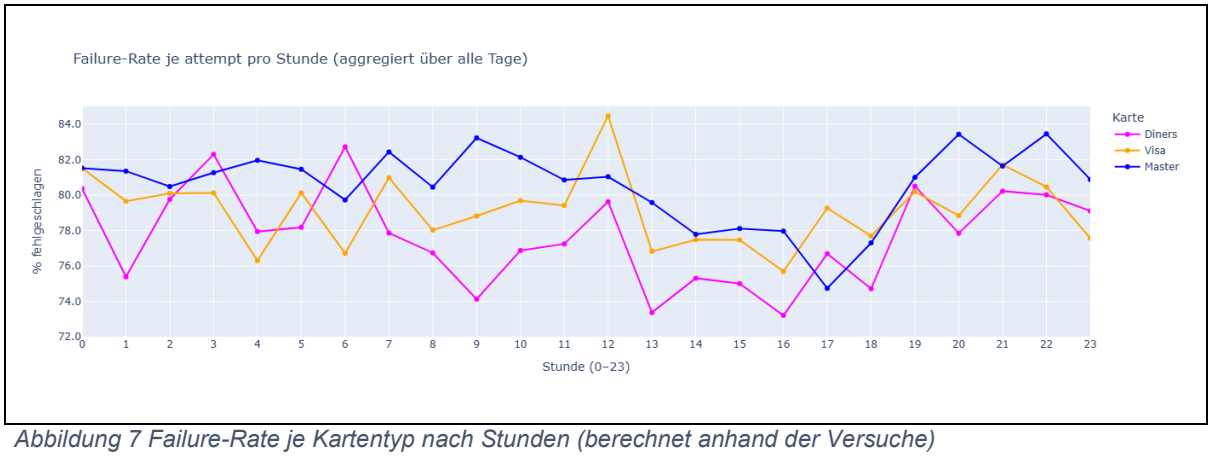

Abbildung 7 Failure-Rate je Kartentyp nach Stunden (berechnet anhand der Versuche)

Die Analyse der stündlichen Fehlerraten je Kartenanbieter (Abbildung 7) zeigt durchgehend hohe und volatile Raten, die oft synchron verlaufen, was auf systemische Ursachen in der Infrastruktur hindeutet. Dabei treten klare stündliche Risikozonen auf: Visa weist eine massive Fehlerspitze um 12 Uhr mittags auf, während Master um 20 und 22 Uhr am anfälligsten ist. Dagegen sind die Raten bei allen zwischen 13 und 19 Uhr am niedrigsten.

	Diners	Master	Visa
Abend	78.6%	81.4%	79.4%
Morgen	77.6%	81.4%	79%
Nachmittag	75.6%	78.2%	78.4%
Nacht	79.1%	81.3%	79.7%

Tabelle 2 Failure Rate nach Kartentyp und Tageszeit (berechnet anhand der Versuche)

Die aggregierten Daten nach Tageszeit (Tabelle 2) bestätigen diese Tendenzen. Die Master-Karte weist über alle Tageszeiten hinweg konstant die höchste Fehlerrate auf (zwischen 81,3 % und 81,4 %), während die Diners-Karte am Nachmittag mit 75,6 % die beste Performance zeigt.

	Diners	Master	Visa
Montag	76.0%	81.1%	78.9%
Dienstag	76.7%	82.6%	81.2%
Mittwoch	78.3%	81.5%	80.0%
Donnerstag	78.6%	81.0%	78.4%
Freitag	79.3%	80.7%	78.8%
Samstag	75.8%	78.7%	76.5%
Sonntag	77.2%	76.6%	78.3%

Tabelle 3 Failure Rate nach Kartentyp und Wochentag (berechnet anhand der Versuche)

Die wöchentliche Betrachtung (Tabelle 3) offenbart ebenfalls Muster. Der Samstag erweist sich für alle Kartenarten als der Tag mit den durchweg niedrigsten Fehlerraten (Diners: 75,8 %; Master: 78,7 %; Visa: 76,5 %). Demgegenüber ist der Dienstag für Master (82,6 %) und Visa (81,2 %) sowie der Freitag für Diners (79,3 %) der Tag mit den höchsten Ausfallquoten.

Analyse nach Zahlungsdienstleister (PSP)

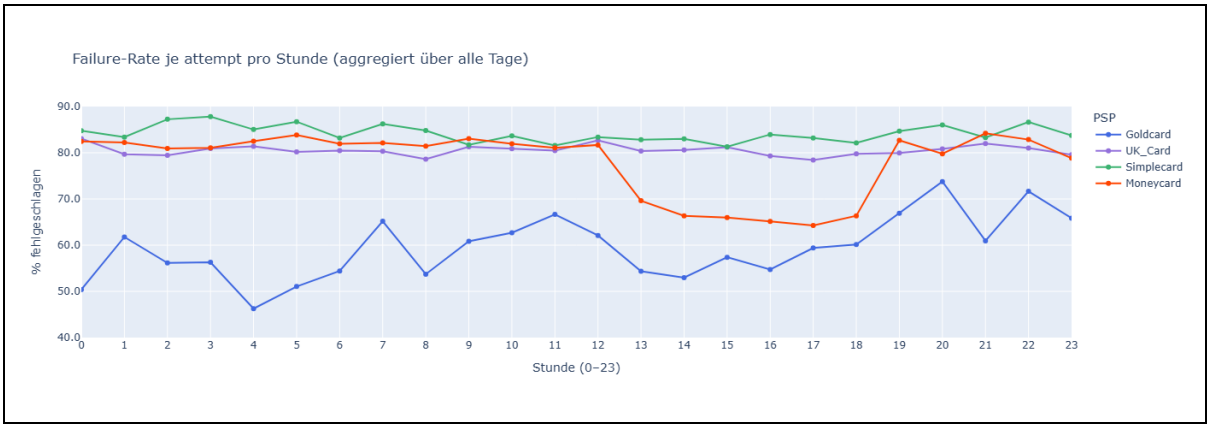

Abbildung 8 Failure-Rate pro Stunde pro PSP (berechnet anhand der Versuche)

Noch prägnanter sind die Profile der PSPs. Die stündliche Analyse (Abbildung 8) offenbart eine klare Zwei-Klassen-Gesellschaft: Goldcard operiert mit einer niedrigeren Fehlerrate (meist 40-70 %) als die drei anderen Anbieter, deren Raten konstant hoch zwischen 70 % und 90 % liegen.

	Goldcard	Moneycard	Simplecard	UK Card
Abend	66.5%	79.3%	84.4%	80.5%
Morgen	60.8%	81.9%	83.6%	80.3%
Nachmittag	56.7%	**68.9%**	82.9%	80.4%
Nacht	53.5%	82.2%	85.8%	80.8%

Tabelle 3 Failure Rate nach PSP und Tageszeit (berechnet anhand der Versuche)

Jeder PSP zeigt dabei eine Persönlichkeit in Abhängigkeit von der Tageszeit, welche in Tabelle 4 weiter untermauert wird. So erweist sich Goldcard nachts als am erfolgreichsten, mit einer Fehlerrate von nur 53,5 %, während sie abends auf 66,5 % ansteigt. Moneycard sticht als ‚Nachmittags-Spezialist' hervor, da die Fehlerrate in diesem Zeitraum von sonst über 81 % auf 68,9 % sinkt. UK Card weist hingegen eine konstant hohe Fehlerrate von etwa 80 % über den gesamten Tag hinweg auf, während Simplecard durchgehend der Anbieter mit der höchsten Fehlerquote (über 82,9 %) ist.

	Goldcard	Moneycard	Simplecard	UK Card
Montag	58.9%	78.0%	82.3%	81.9%
Dienstag	62.6%	80.0%	86.1%	81.6%
Mittwoch	62.1%	79.9%	85.3%	80.7%
Donnerstag	57.5%	77.5%	82.8%	82.2%
Freitag	57.0%	78.4%	85.3%	80.7%
Samstag	60.7%	75.7%	83.5%	77.7%
Sonntag	53.8%	75.4%	82.7%	78.4%

Tabelle 4 Failure Rate nach PSP und Wochentag (berechnet anhand der Versuche)

Die wöchentliche Analyse in Tabelle 5 bestätigt diese Profile. Goldcard erreicht am Sonntag mit 53,8 % die niedrigste wöchentliche Fehlerrate aller Anbieter, während sie am Dienstag mit 62,6 % am höchsten ist. Simplecard weist am Dienstag mit 86,1 % den höchsten wöchentlichen Fehlerwert auf. Moneycard und UK Card zeigen ebenfalls Schwankungen, wobei deren Fehlerraten am Wochenende (Samstag und Sonntag) tendenziell am niedrigsten sind.

Analyse kombinierter Merkmale und deren Einfluss

Die stärksten Prädiktoren für den Transaktionserfolg werden erst durch die Kombination von Merkmalen heraus.

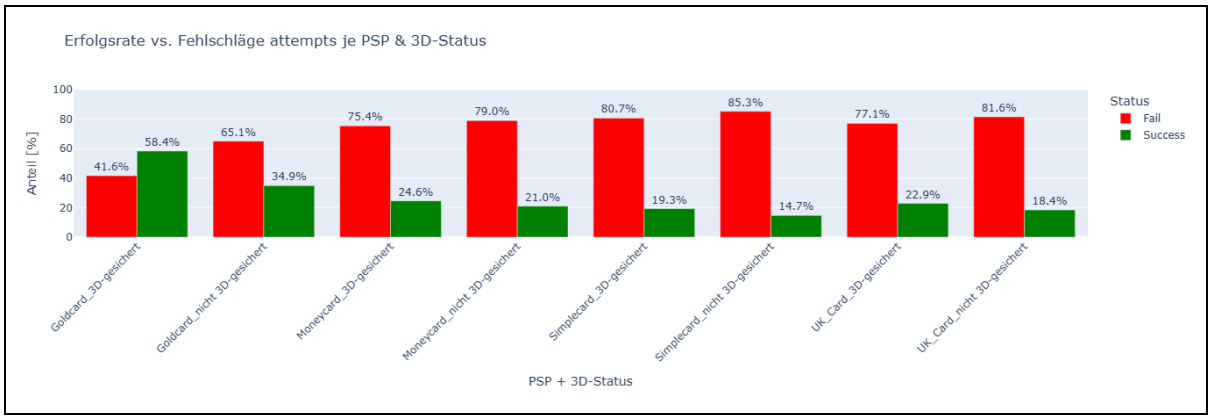

Abbildung 9 Prozentualer Anteil von PSP & 3D-Status Kombinationen (berechnet anhand der Versuche und aufgeteilt in Success und Fail)

Die Verwendung von 3D-Secure hat einen starken Einfluss, insbesondere bei Goldcard, wo die Erfolgsrate mit 3D-Secure von 34,9 % auf 58,4 % ansteigt (Abbildung 9).

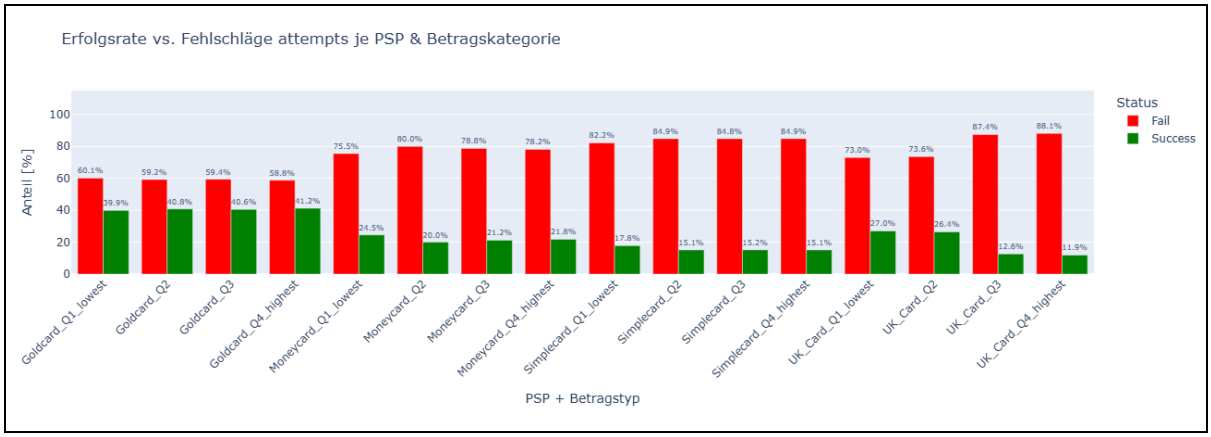

Abbildung 10 Prozentualer Anteil von PSP & Betragskategorie Kombinationen (berechnet anhand der Versuche und aufgeteilt in Success und Fail)

Ebenso ist die Betragshöhe entscheidend: Bei UK Card steigt die Fehlerrate für hohe Beträge dabei auf bis zu 88,1 %, während sie bei niedrigen Beträgen bei ca. 73 % liegt (Abbildung 10).

Korrelationsanalyse zur Quantifizierung der Zusammenhänge

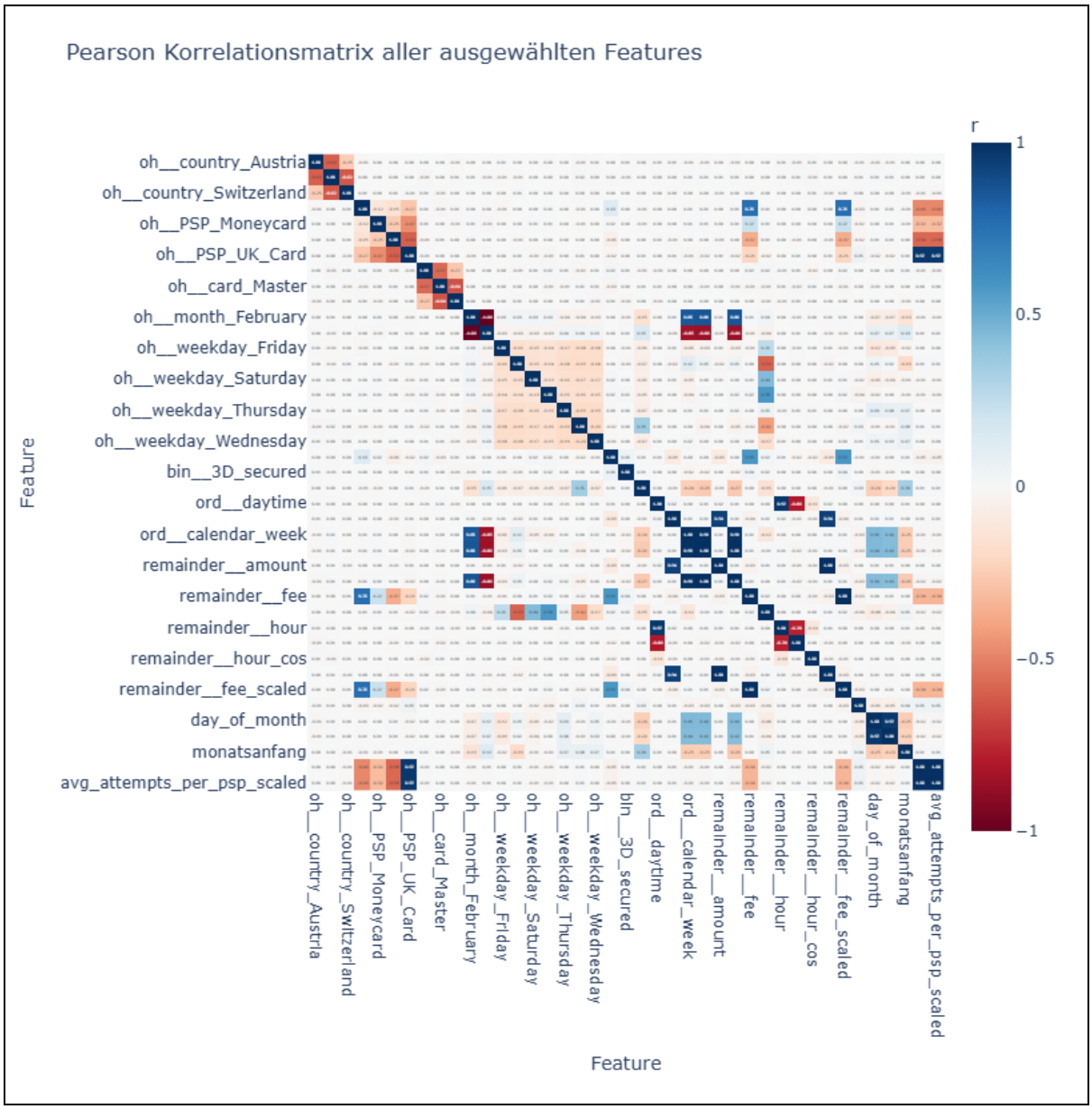

Abbildung 11 Pearson Korrelationsmatrix aller ausgewählten Features

Zur weiteren Untersuchung dieser Beobachtungen wurde eine Korrelationsanalyse durchgeführt, die lineare Zusammenhänge (Pearson) zwischen den Merkmalen misst (Soetewey, 2023). Die Pearson-Korrelationsmatrix aller ausgewählten Features zeigt, dass die Gebühr (remainder__fee_scaled) die höchste positive Korrelation zum Transaktionserfolg aufweist (r ≈ 0.59). Allerdings stellt dies ein Leakage-Problem dar, da Gebühren direkt aus erfolgreichen Transaktionen resultieren, weshalb dieses Merkmal für Vorhersagemodelle ungeeignet ist. Die Nutzung der Goldcard zeigt mit r ≈ 0.13 die höchste relevante positive Korrelation. Negativ korrelieren hingegen der Transaktionsbetrag (r ≈ -0.09) und die Nutzung der Simplecard (r ≈ -0.06).

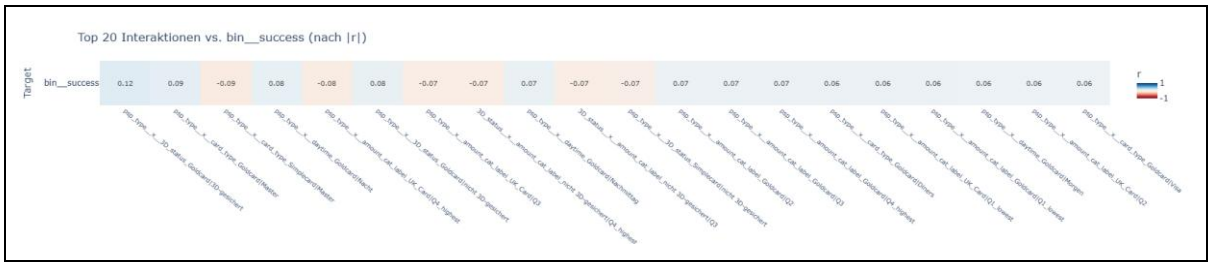

Abbildung 12 Pearson Korrelationsmatrix aller Kreuzvariablen mit Zielvariable success

Die Analyse der Kreuzvariablen bestätigt diese Ergebnisse (siehe Abbildung 12). So weist beispielsweise die Kombination aus Goldcard und 3D-Secure mit einer Korrelation von r ≈ 0.12 den stärksten positiven Zusammenhang mit dem Transaktionserfolg auf, gefolgt von der Verbindung zwischen Goldcard und Mastercard (r ≈ 0.09). Im Gegensatz dazu zeigen die Kombination von Simplecard mit Mastercard (r ≈ -0.09) sowie die Verbindung von UK Card mit hohen Beträgen (r ≈ -0.08) negative Korrelationen. Eine segmentierte Korrelationsanalyse nach Tageszeit unterstreicht zudem die Sonderrolle bestimmter Anbieter zu spezifischen Zeiten. Insbesondere die positive Korrelation von Goldcard in der Nacht (r ≈ 0.177) und Moneycard am Nachmittag (r ≈ 0.111) sind zeitlich erkennbare Muster. Im Gegensatz dazu deckt die Wochentags Segmentierung keine markanten Abweichungen von den globalen Mustern auf. Um diese potenziell prädiktiven Muster für ein nachfolgendes Machine-Learning-Modell explizit hervorzuheben, wurden die neuen, binär kodierten Spalten bin__daytime_Nacht und bin__daytime_Nachmittag erstellt. Diese fungieren als Indikatorvariablen, die den Wert 1 annehmen, wenn eine Transaktion in der Nacht bzw. am Nachmittag stattfindet, und andernfalls den Wert 0.

4.3.2 EDA für die Zielvariable Fee

Im Rahmen der explorativen Datenanalyse wird nachfolgend die Zielvariable Fee untersucht, um die wesentlichen Einflussfaktoren und Kostenstrukturen im Zahlungsprozess zu identifizieren.

Fee je Karten und PSP Typ

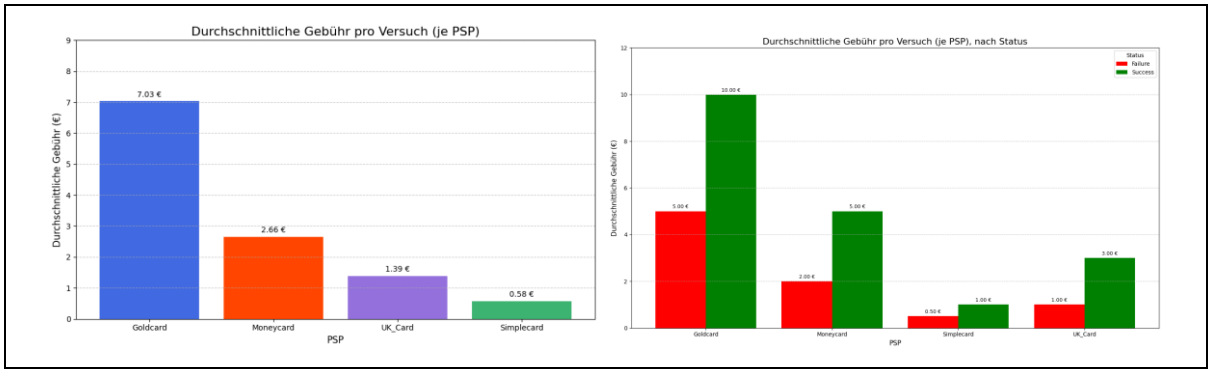

Abbildung 13 Durchschnittliche Gebühr je PSP und Kartentyp (berechnet anhand der Versuche und aufgeteilt in Success und Fail)

Die Analyse der durchschnittlichen Kosten pro Versuch spiegelt exakt die erwartete Gebührenhierarchie gemäß dem PSP-Kostenregelwerk wider, wobei Goldcard (durchschnittlich 7,03 €) am teuersten und Simplecard (0,58 €) am günstigsten ist (Abbildung 13). Die Aufschlüsselung nach Status bestätigt dieses Bild: erfolgreiche Transaktionen sind konsequent teurer als fehlgeschlagene Versuche (z.B. Goldcard: 10,00 € bei Erfolg vs. 5,00 € bei Fehlschlag), was exakt den im Regelwerk definierten unterschiedlichen Abrechnungsmodi für erfolgreiche und abgelehnte Buchungen entspricht.

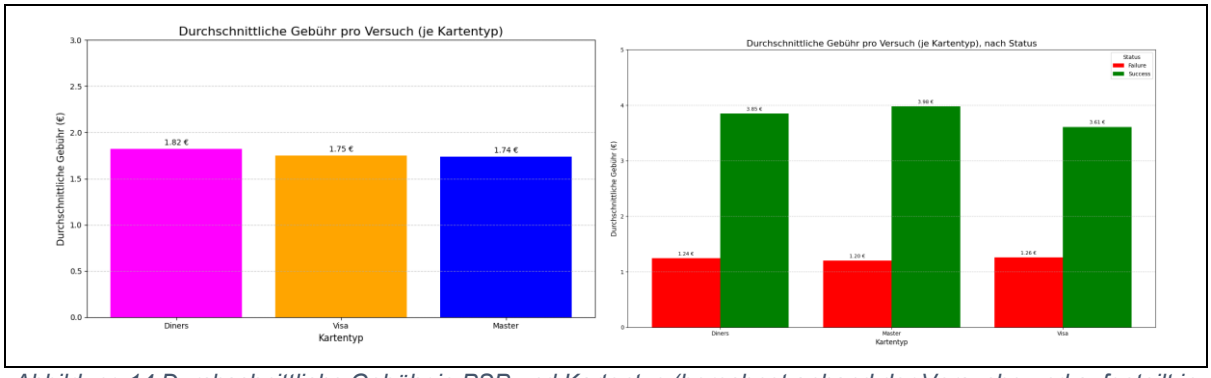

Abbildung 14 Durchschnittliche Gebühr je PSP und Kartentyp (berechnet anhand der Versuche und aufgeteilt in Success und Fail)

Im klaren Gegensatz zu den PSPs zeigt die Analyse der Kartentypen nur minimale Kostenunterschiede. Die durchschnittlichen Gebühren pro Versuch liegen mit 1,82€ (Diners), 1,75€ (Visa) und 1,74€ (Master) nahezu gleichauf (Abbildung 14). Dieses konsistente Bild setzt sich bei der Aufteilung nach Status fort, wobei die Kosten für Misserfolge (ca. 1,20€ -

1,26€) und Erfolge (ca. 3,61€ - 3,98€) über alle drei Kartentypen hinweg sehr ähnlich strukturiert sind.

Der Zusammenhang zwischen Gebühr und Transaktionserfolg

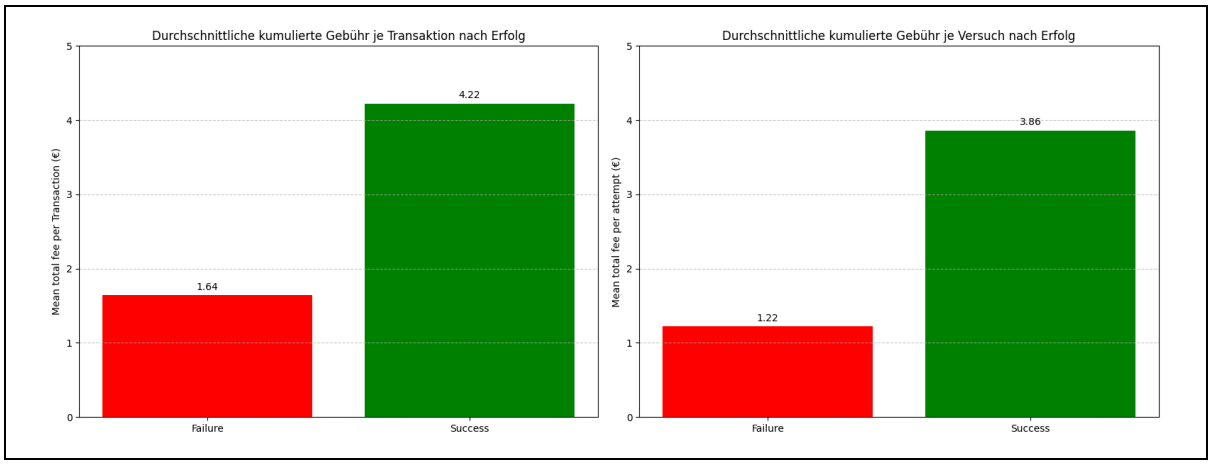

Abbildung 15 Durchschnittliche kumulierte Gebühr je Transaktion und Versuch (aufgeteilt in Success und Fail)

Wie in der Abbildung 15 visualisiert, beläuft sich die durchschnittliche Gesamtgebühr für eine erfolgreiche Transaktion auf 4,22 €, während eine vollständig fehlgeschlagene Transaktion lediglich Kosten von 1,64 € verursacht. Dieses Muster bestätigt sich auch auf der Ebene der einzelnen Versuche, wo ein erfolgreicher Versuch im Schnitt 3,86 € kostet, ein fehlgeschlagener hingegen nur 1,22 €. Die Ursache hierfür liegt bei an den Kosten des Erfolgs an sich, der teurer ist als ein Abbruch (Tabelle 1).

Kosteneskalation in Abhängigkeit der Versuchszahl

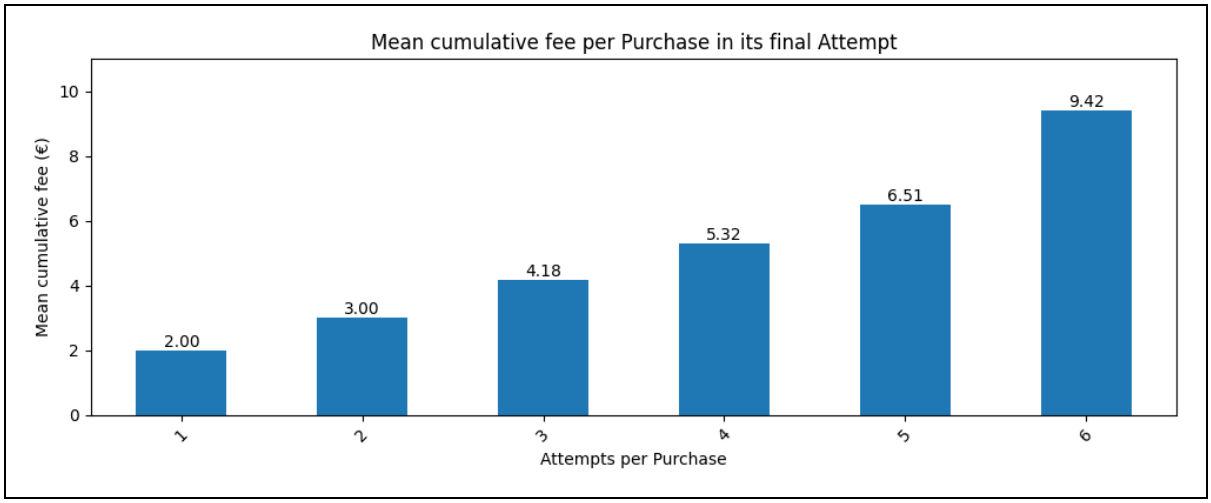

Abbildung 16 Durchschnittliche kumulierte Gebühr pro Transaktion, aufgeschlüsselt nach der Anzahl der Versuche

Abbildung 16 belegt zudem eine direkte Korrelation zwischen der Anzahl der Versuche und den Gesamtkosten einer Transaktion. Da jeder Versuch Kosten verursacht, führt demnach auch jeder weitere Durchgang zu höheren Transaktionskosten. Eine Transaktion, die im ersten Anlauf erfolgreich ist, verursacht im Durchschnitt Gebühren von 2,00 €, wohingegen

eine Transaktion, die sechs Versuche benötigt, durchschnittliche Gesamtkosten von 9,42 € erreicht. Dies verdeutlicht, dass die Komplexität und der anfängliche Misserfolg der Abwicklung die primären Kostentreiber sind.

Einfluss weiterer Merkmale auf die Gebührenstruktur

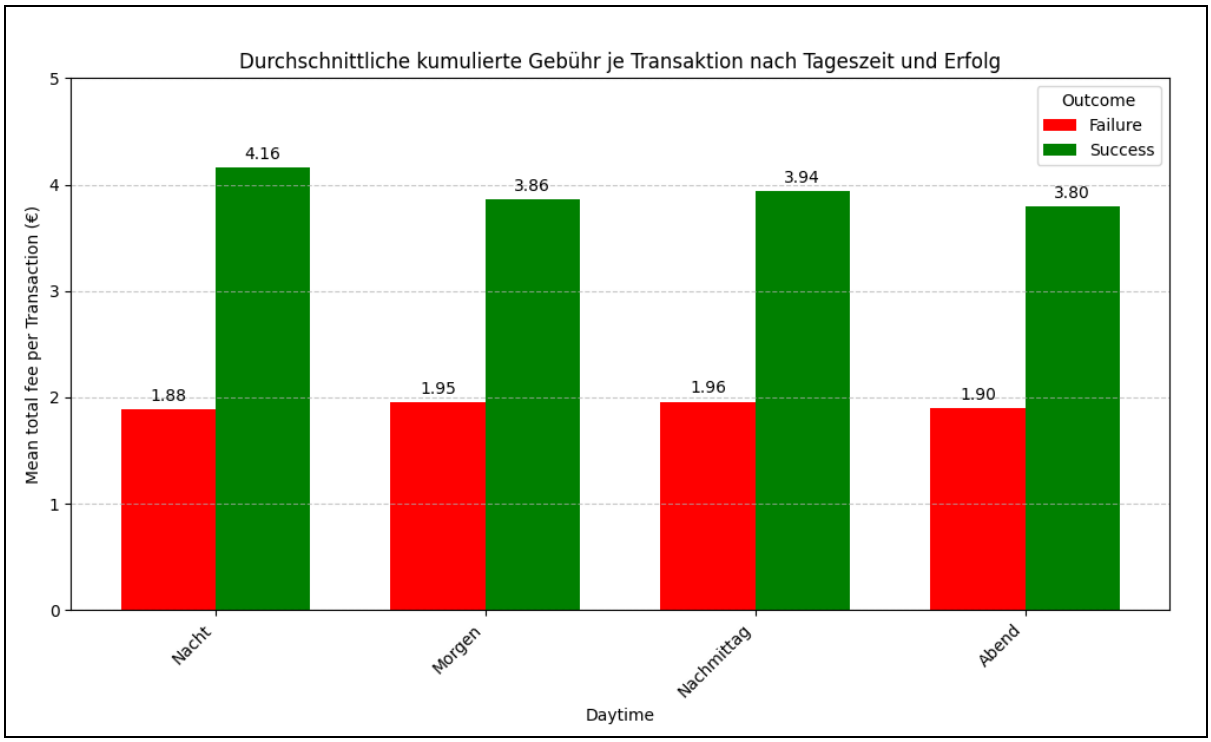

Abbildung 17 Durchschnittliche kumulierte Gebühr nach Tageszeit (berechnet anhand der Transaktion und aufgeteilt in Success und Fail)

Die Tageszeit beeinflusst die Höhe der Gebühren für erfolgreiche Transaktionen, da die höchsten Durchschnittsgebühren (4,16 €) in der Nacht anfallen (Abbildung 17). Die Kosten für fehlgeschlagene Transaktionen bleiben hingegen über den Tag hinweg sehr stabil bei etwa 1,90 €.

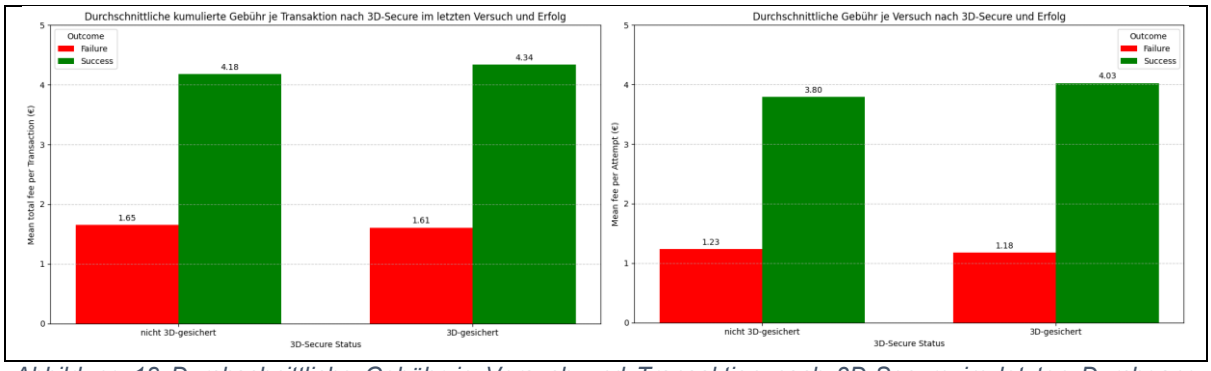

Abbildung 18 Durchschnittliche Gebühr je Versuch und Transaktion nach 3D-Secure im letzten Durchgang (aufgeteilt in Success und Fail)

Auch der 3D-Secure-Status spielt eine Rolle (Abbildung 18). Die höchste durchschnittliche kumulierte Gebühr von 4,34 € wird bei erfolgreichen Transaktionen verzeichnet, deren letzter

Versuch 3D-Secure-gesichert war. Auf Versuchsebene ist ein erfolgreicher, 3D-gesicherter Versuch mit 4,03 € am teuersten.

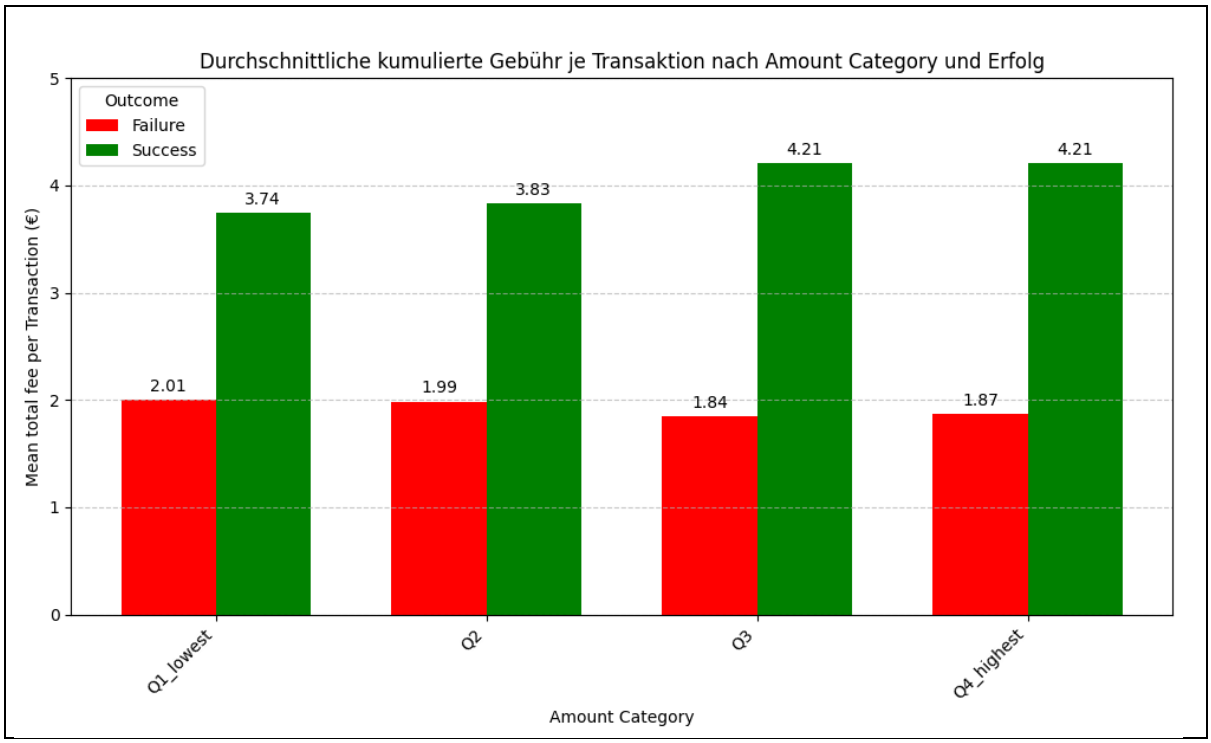

Abbildung 19 Durchschnittliche kumulierte Gebühr nach Betragskategorie (berechnet anhand der Transaktion und aufgeteilt in Success und Fail)

Die Betragskategorie zeigt ebenfalls einen klaren Zusammenhang (Abbildung 19). Während die Kosten für fehlgeschlagene Transaktionen stabil bleiben, steigen die Gebühren für erfolgreiche Transaktionen mit dem Betrag an und erreichen in den höchsten Kategorien (Q3 und Q4) einen Wert von 4,21 €. Dies ist darauf zurückzuführen, dass Transaktionen mit höheren Beträgen häufiger fehlschlagen und somit mehr Versuche benötigen, was die kumulierten Kosten in die Höhe treibt.

Korrelationsanalyse und finale Feature-Auswahl

Die Korrelationsanalyse quantifiziert diese Zusammenhänge und verdeutlicht, dass die Gebührenhöhe primär von der Wahl des PSPs abhängt (Abbildung 11). Die Nutzung der Goldcard weist eine sehr starke positive Korrelation ($r \approx 0.758$) mit den Gebühren auf, während die Simplecard stark negativ korreliert ($r \approx -0.372$), was bestätigt, dass diese Anbieter die Gebührenstruktur maßgeblich definieren.

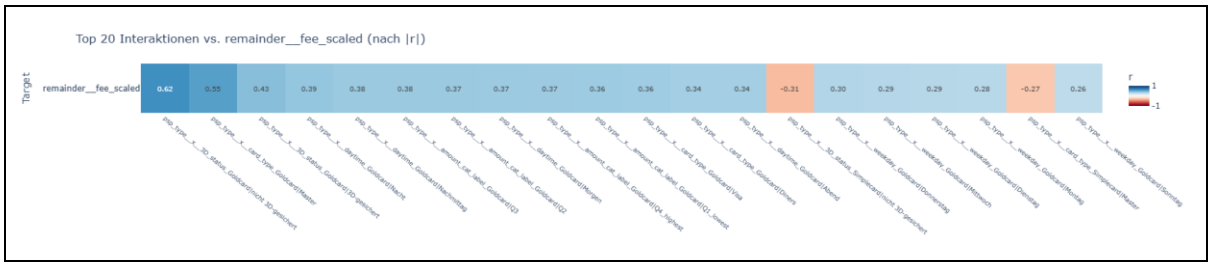

Abbildung 20 Pearson Korrelationsmatrix aller Kreuzvariablen mit Zielvariable Fee

Die Analyse von Interaktionsvariablen untermauert dies: Die Kombination "Goldcard × nicht 3D-gesichert" weist mit r = +0,618 die stärkste positive Korrelation zu hohen Gebühren auf. Das hängt mit der höheren Erfolgschance von Goldcard in Kombination mit 3D Sicherungen und der teuersten Fee von 10 Euro bei erfolgreichen GoldCard Transaktionen zusammen. Wie bei der Zielvariable Fee wurden die Korrelationen je PSP in Kombination mit den Wochentagen und den Tageszeiten zur Zielvariable gemessen. Die Wochentagssegmentierung fügt hier keine neue Information hinzu, sondern bestätigt, dass PSP-Typ der primäre, dominante Faktor für die Gebührenhöhe ist.

4.3.3 Fazit der Explorativen Datenanalyse

Für die finale Modellerstellung wurden schließlich zahlreiche Spalten entfernt, um Multikollinearität, Data Leakage (z.B. attempt) und geringe Informationsdichte zu vermeiden. Die finale Feature-Liste konzentriert sich auf die aussagekräftigsten Prädiktoren: `bin__success`, `oh__PSP_Goldcard`, `oh__PSP_Moneycard`, `oh__PSP_Simplecard`, `oh__PSP_UK Card`, `oh__card_Diners`, `oh__card_Master`, `oh__card_Visa`, `remainder__amount_scaled`, `remainder__fee`, `bin__daytime_Nacht`, `bin__daytime_Nachmittag`, `remainder__hour_sin`, `bin__3D_secured`, `ord__amount_cat`, `monatsanfang`. Dabei werden `remainder__fee` und `bin__success` jeweils aus Leakage Gründen aus den Dataframes der Klassifikation und der Regression entfernt.

Zusammenfassend hat die explorative Datenanalyse gezeigt, dass die Vorhersage der Zielvariable Erfolg komplex ist, da kein einzelnes Merkmal eine überragend starke prädiktive Kraft besitzt. Insbesondere die linearen Korrelationen einzelner Merkmale mit dem Transaktionserfolg fielen gering aus, was darauf hindeutet, dass einfache lineare Modelle hier an ihre Grenzen stoßen würden. Aussagekräftige Muster entstanden erst durch die Untersuchung von Interaktionen zwischen verschiedenen Variablen, wie beispielsweise der Kombination aus dem Zahlungsdienstleister (PSP), dem 3D-Secure-Status, der Betragshöhe und der Tageszeit. Im deutlichen Gegensatz dazu zeigte die Analyse der Zielvariable Fee wesentlich stärkere und direktere Zusammenhänge. Hier erwies sich der gewählte Zahlungsdienstleister als dominanter Faktor mit sehr hohen linearen Korrelationen, wie die

starke positive Korrelation der Goldcard ($r \approx 0.76$) und die moderate negative Korrelation der Simplecard ($r \approx -0.37$) belegen. Diese Erkenntnisse legen nahe, dass für die Vorhersage des Transaktionserfolgs ein Modell erforderlich ist, das komplexe Interaktionen erfassen kann, während die Gebührenvorhersage durch einfachere, direktere Beziehungen geprägt ist.

5 Modellierung

5.1 Feature Selektion

Die Merkmalsauswahl, auch bekannt als Feature Selection, ist ein entscheidender Prozess in der Entwicklung von Machine-Learning-Modellen. Bei diesem Vorgehen wird gezielt eine Untermenge der relevantesten Merkmale aus dem gesamten Datensatz ausgewählt. Das primäre Ziel besteht darin, die Modellleistung zu verbessern, indem irrelevante oder redundante Features entfernt werden. Dieses Vorgehen reduziert das Risiko von Overfitting, verkürzt die Trainingszeiten und führt oft zu einfacheren und besser interpretierbaren Modellen.

Basierend auf den Prinzipien des Feature Engineerings und der Dimensionsreduktion wurden in diesem Schritt mehrere finale Daten-Versionen für die nachfolgende Modellevaluierung erstellt, die jeweils eine andere Strategie der Merkmalsauswahl repräsentieren. Als Ausgangspunkt dienten zwei aufbereitete Datensätze: `df_Classification` für die Vorhersage des Transaktionserfolgs (`bin__success`) und `df_Regression` für die Vorhersage der Transaktionsgebühr (`remainder__fee`). Für beide Aufgaben wurden zwei Hauptstrategien verfolgt: die Hauptkomponentenanalyse (PCA), um die Dimensionalität zu reduzieren und die Informationen der ursprünglichen Features in unkorrelierten Hauptkomponenten zu komprimieren, sowie diverse Filter- und Wrapper-Methoden wie SelectKBest, SequentialFeatureSelector (SFS) und Recursive Feature Elimination (RFE) (Werk 1: scikit-learn, n.d.; Werk 2: scikit-learn, n.d.) geeksforgeeks, 2025 (geeksforgeeks, 2025). Aus diesen Methoden wurden für jede der beiden Aufgaben spezifische DataFrames generiert, um deren Leistung im späteren Modellvergleich evaluieren zu können. Als Ergebnis dieses umfassenden Prozesses wurden für jede der beiden Aufgaben – Klassifikation und Regression – vier spezifische DataFrames für die Modellevaluierung erstellt und gespeichert. Neben den vollständigen Referenz-DataFrames (`df_Classification`, `df_Regression`) existieren nun die PCA-Original-Features-DataFrames, die das Subset der als am relevantesten identifizierten Original-Features umfassen (`df_classification_pca_original_features`, `df_Regression_pca_original_features`). Des Weiteren wurden die Feature-Selection-DataFrames (`df_classification_featureSelection`, `df_Regression_featureSelection`) aus der Vereinigung der Ergebnisse der Filter- und

Wrapper-Methoden gebildet. Schließlich wurden die `Reduced-DataFrames` (`df_classification_reduced`, `df_Regression_reduced`) erstellt, die eine Kombination aus der Feature-Selektion und den PCA-basierten Original-Features darstellen, um eine möglichst umfassende und dennoch reduzierte Merkmalsmenge zu erhalten. Die genauen Spalten je Dataframe und Verfahren können im File „Feature_Selection.ipynb“ betrachtet werden.

5.2 Modell Training

Nach der Feature-Selektion beginnt das Modell-Training. Das Ziel ist die Entwicklung von Vorhersagemodellen, die sowohl die Erfolgswahrscheinlichkeit als Klassifikation als auch die voraussichtlichen Kosten als Regression präzise abbilden.

5.2.1 Klassifikation

Im praktischen Einsatz analysiert das Klassifikationsmodell eine Transaktion und berechnet für jeden der vier verfügbaren Zahlungsdienstleister eine separate Erfolgswahrscheinlichkeit, um eine datengestützte Routing-Entscheidung zu ermöglichen. In diesem Kapitel wird demnach ein Klassifikationsmodell entwickelt, welches die primäre Aufgabe hat, möglichst gute Metriken darin zu erzielen je PSP den Erfolg vorauszusagen.

Evaluationsmetriken und strategische Ausrichtung

Die wichtigsten Metriken des Modells sind stark von den Wünschen des Unternehmens abhängig. Dabei liegt der Fokus laut den Zusatzinformationen darin die Kundenzufriedenheit zu steigern, was bedeutet die Abbruchquote möglichst gering zu halten.

	Vorhersage: Abbruch (0)	**Vorhersage: Erfolg (1)**
Wahrheit: Abbruch (0)	**True Negative (TN)** Wahr Negativ. **Erklärung**: Modell sagt korrekt Abbruch voraus. **Geschäftsauswirkung**: Fehlversuch wurde vermieden.	**False Positive (FP)** Falsch Positiv (Fehler 1. Art). **Erklärung**: Modell sagt fälschlicherweise Erfolg voraus. **Geschäftsauswirkung**: Führt zu fehlgeschlagenem Versuch und unzufriedenem Kunden.
Wahrheit: Erfolg (1)	**False Negative (FN)** Falsch Negativ (Fehler 2. Art). **Erklärung**: Modell sagt fälschlicherweise Abbruch voraus. **Geschäftsauswirkung**: Ein Versuch, der erfolgreich gewesen wäre, wurde nicht durchgeführt.	**True Positive (TP)** Wahr Positiv. **Erklärung**: Modell sagt korrekt Erfolg voraus. **Geschäftsauswirkung**: Ein Versuch wurde erfolgreich durchgeführt.

Tabelle 5 Konfusionsmatrix

Die zugrundeliegende Konfusionsmatrix (wie in Tabelle 5 dargestellt) dient als Basis für diese Metriken und verdeutlicht die unterschiedlichen Kosten der Fehlklassifikationen: Ein Falsch Positiver Fall, bei dem das Modell fälschlicherweise einen Erfolg vorhersagt, obwohl die Realität ein Abbruch ist, stellt den schwerwiegendsten Fehler dar. Dies liegt daran, dass ein FP zu einem fehlgeschlagenen Zahlungsversuch, direkten Kosten und einem unzufriedenen Kunden führt. Um diesen kritischen Fehler zu minimieren, wurde ein starker Fokus auf eine hohe Precision gelegt, die den Anteil der echten Erfolge (TP) an allen als erfolgreich vorhergesagten Fällen misst. Gleichzeitig ist ein hoher Recall essenziell, um sicherzustellen, dass das Modell möglichst viele der tatsächlich erfolgreichen Transaktionen identifiziert und somit Umsatzchancen nicht ungenutzt lässt. Eine hohe Precision und ein niedriger Recall würden bedeuten, dass Erfolge zu streng akzeptiert werden, weshalb fast alle tatsächlichen Erfolge als Abbruch kategorisiert werden. Als primäres Ziel wurde daher eine hohe Balance zwischen Precision und Recall definiert, mit einer leichten Priorisierung der Precision aufgrund der hohen Kosten eines False Positives. Die Specificity misst den Anteil der korrekt identifizierten misslungenen Transaktionen (TN) an allen tatsächlich misslungenen Fällen und ist aufgrund der Häufigkeit von Fehlversuchen ebenfalls relevant für die Bewertung der Modellgüte. Die AUC (Area Under the Curve) diente als schwellenwertunabhängige Hauptmetrik zur Beurteilung der allgemeinen Vorhersagekraft des Modells, insbesondere im Kontext der stark unausgeglichenen Klassenzugehörigkeit, während der F1-Score als harmonisches Mittel zur Bewertung des Kompromisses zwischen Precision und Recall herangezogen wurde. Der Metrik Accuracy (Genauigkeit) wird hingegen eine untergeordnete Bedeutung beigemessen. Angesichts der starken Klassen-Imbalance im Datensatz wäre diese Kennzahl irreführend, da bereits ein naives Modell, das konsequent die Mehrheitsklasse („Fail“) vorhersagt, eine künstlich hohe Genauigkeit (~80%) erzielen würde.

Baseline-Modell und Auswahl des Datensatzes

Als erster Schritt wurde ein robustes Baseline-Modell mittels logistischer Regression etabliert, um einen grundlegenden Leistungs-Benchmark zu definieren. Die Baseline wurde auf den verschiedenen, im vorherigen Kapitel erstellten Datensätzen trainiert und evaluiert. Die Ergebnisse zeigten, dass alle Feature-Sets eine nahezu identische Performance lieferten. Aufgrund dessen wurde der Datensatz df_classification_reduced für die weitere Modellierung ausgewählt, da er bei fast identischer Leistung die effizienteste Grundlage bot. Das Baseline-Modell selbst offenbarte die Herausforderung des stark unbalancierten Klassifikationsproblems: Obwohl eine hohe Accuracy (0.798) und Specificity (0.997) erreicht wurden und die Precision bei 0.55 lag, waren der Recall (0.018) und der F1-Score (0.034) extrem niedrig, was zeigte, dass das Modell die seltene Erfolgsklasse kaum erkannte und demnach fast immer die Mehrheitsklasse voraussagt.

Evaluierung und Vergleich anspruchsvollerer Modelle

Nachdem die initiale Baseline etabliert war, wurde eine breite Palette von Algorithmen trainiert und evaluiert, darunter baumbasierte Ensembles wie Random Forest und XGBoost sowie ein Gaussian Mixture Model (GMM). Für jedes dieser Modelle erfolgte eine automatisierte Hyperparameter-Optimierung mittels Hyperopt, um die jeweils bestmögliche Konfiguration zu ermitteln. Abschließend wurde im MLflow-Tracking für jeden Algorithmus das Modell mit der Balance aus dem besten AUC-Score, der besten Precision und dem besten Recall identifiziert und als PKL-Datei gespeichert.

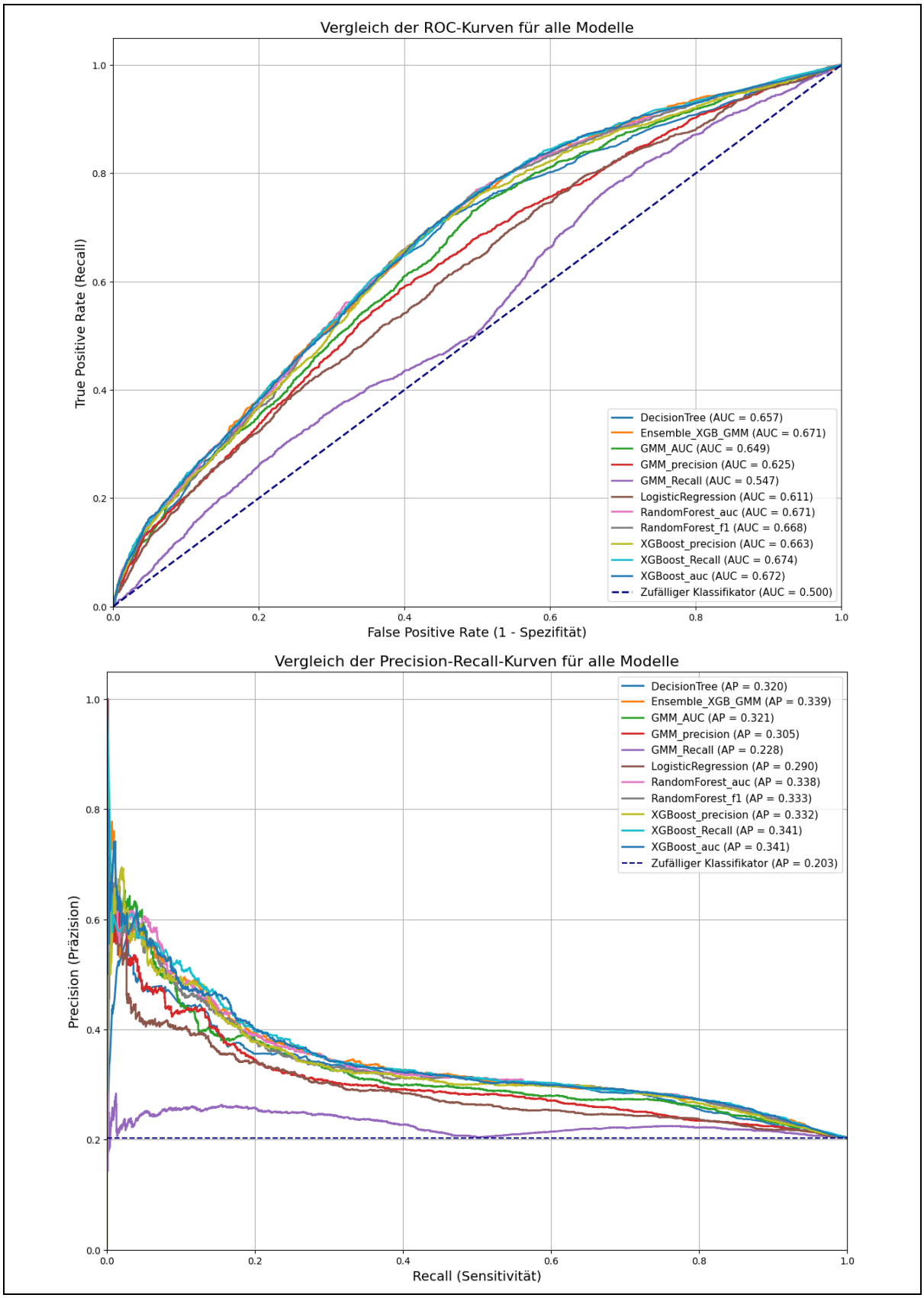

Abbildung 21 Vergleich der ROC-Kurven und Vergleich der Precision-Recall-Kurven für alle Modelle

Die vergleichende Analyse der ROC- und Precision-Recall-Kurven (Abbildung 20) zeigte eine klare Überlegenheit der baumbasierten Ensemble-Methoden. Die XGBoost- und

RandomForest-Modelle dominierten die Spitze mit den höchsten AUC-Werten (bis zu 0.674) und Average Precision (AP) Scores (bis zu 0.341), was ihre Fähigkeit unterstreicht, die komplexen, nicht-linearen Muster in den Daten am besten zu erfassen. Im Gegensatz dazu zeigten die GMM-Varianten, die auf unterschiedliche Metriken wie AUC, Precision oder Recall optimiert wurden, eine insgesamt schwächere oder extrem unausgewogene Leistung. Basierend auf diesen Ergebnissen wurde das auf AUC optimierte XGBoost-Modell (XGBoost_auc) als leistungsstärkster Kandidat für die weitere, tiefere Analyse ausgewählt.

Segmentierte Analyse und strategische Threshold-Optimierung des XGBoost-Modells

Eine detaillierte Analyse des ausgewählten XGBoost-Modells auf PSP-Ebene offenbarte starke Leistungsunterschiede zwischen den Segmenten.

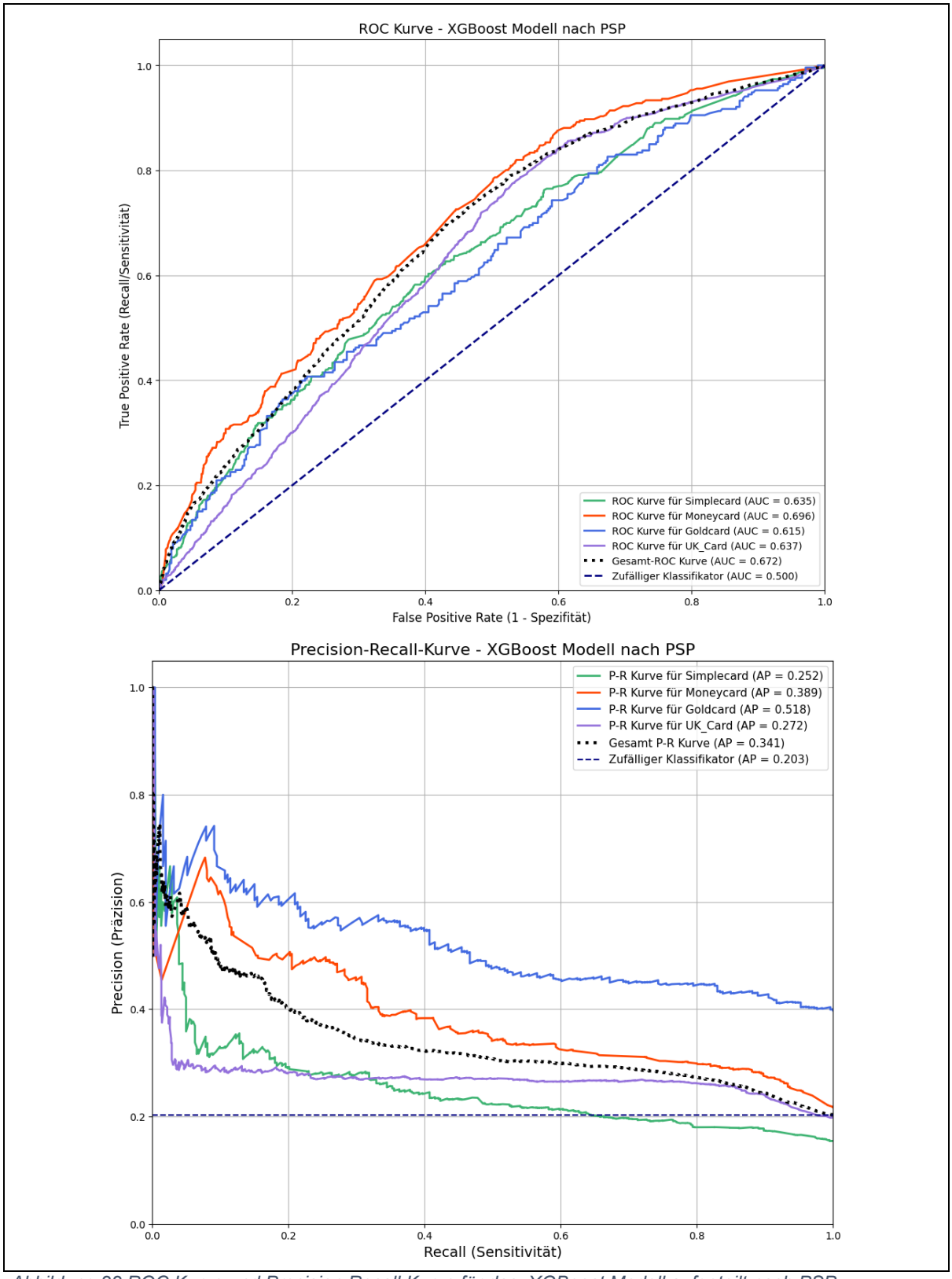

Abbildung 22 ROC Kurve und Precision Recall Kurve für das XGBoost Modell aufgeteilt nach PSP

Die ROC-Kurve nach PSP (Abbildung 22) zeigte, dass das Modell für Moneycard (AUC = 0.696) die höchste Unterscheidungsfähigkeit aufwies, während es für Goldcard (AUC = 0.615) am schwächsten performte. Noch deutlicher wurden diese Unterschiede in der Precision-Recall-Kurve, wo Transaktionen über Goldcard (AP = 0.518) am besten zu klassifizieren waren, während Simplecard (AP = 0.252) und UK Card (AP = 0.272) nahe am Zufallsniveau lagen. Diese Heterogenität belegte, dass ein globaler Entscheidungsschwellenwert (Threshold) nicht für alle PSPs optimal sein kann.

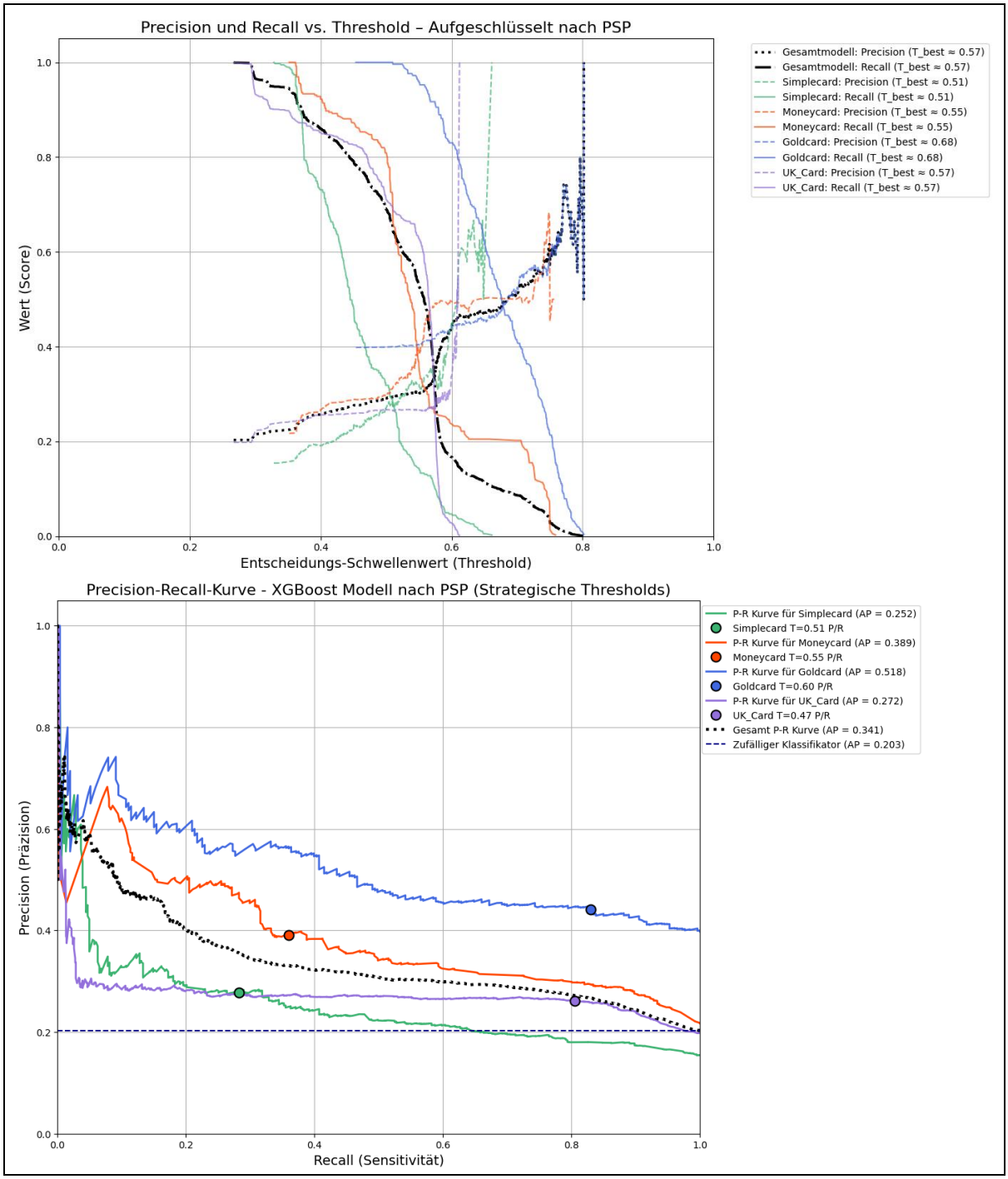

Abbildung 23 Optimale Treshholdfindung des XGBoost Modells - Aufgeschlüsselt nach PSP

Die Analyse der Metriken in Abhängigkeit vom Schwellenwert (Abbildung 23) bestätigte dies. Der optimale Kompromisspunkt zwischen Precision und Recall variierte stark je nach PSP. Dies führte zur Festlegung der finalen, strategischen Thresholds, die in der Precision-Recall-Kurve markiert sind. Die finalen, strategisch angepassten Thresholds lauten: Simplecard (T=0.51), Moneycard (T=0.55), Goldcard (T=0.60) und UK Card (T=0.47).

Finale Modellarchitektur und Leistungsbewertung

Das finale Klassifikationsmodell ist somit der auf AUC optimierte XGBoost Classifier, dessen Vorhersagen jedoch nicht mit einem globalen, sondern mit PSP-spezifischen Schwellenwerten in eine binäre Entscheidung (Erfolg/Misserfolg) überführt werden.

Metrik	Wert
Datensätze im Validierungsset	10066
Accuracy	0.613
F1-Score	0.399
Precision	0.292
Recall	0.632
AUC	0.672

Tabelle 6 XGBoost Modell Metriken

Die Gesamtperformance dieses finalen Ansatzes (Tabelle 6) auf dem Validierungsset mit 10.066 Datensätzen zeigt eine solide Accuracy von 0.613. Die hohe Recall-Rate von 0.632 wird durch eine Precision von 0.292 erkauft, was zu einem F1-Score von 0.399 führt. Entscheidender ist die generelle Trennschärfe des Modells, die durch einen guten Gesamt-AUC-Score von 0.672 bestätigt wird.

PSP	Threshold	AUC	Precision	Recall	F1-Score	Accuracy	TN	FP	FN	TP
Simplecard	0.51	0.634	0.28	0.28	0.28	0.7759	1814	280	275	108
Moneycard	0.55	0.696	0.39	0.36	0.37	0.7389	1099	202	232	129
Goldcard	0.60	0.615	0.44	0.83	0.58	0.5142	117	266	43	210
UK Card	0.47	0.634	0.26	0.81	0.39	0.5097	1852	2390	204	845

Tabelle 7 XGBoost Modell Metriken je PSP

Die segmentierte Analyse (Tabelle 7) offenbart, wie die strategischen Thresholds die Leistung individuell steuern und wie sich die grundlegende Trennschärfe (AUC) je PSP unterscheidet. Moneycard zeigt die stärkste Modellleistung. Die AUC ist mit 0.696 die höchste im Vergleich, was bedeutet, dass das Modell hier am besten zwischen Erfolg und Misserfolg unterscheiden kann. Der gewählte Threshold führt zu einem ausgewogenen F1-Score von 0.37. Simplecard und UK Card weisen eine identische, moderate Trennschärfe auf (AUC 0.634). Bei UK Card wird der Recall durch den niedrigen Threshold aggressiv auf 0.81 maximiert, da die Precision zwischen Recall 0.2 und 0.81 stagniert (Abbildung 23). Goldcard (T=0.60) erzielt ebenfalls einen sehr hohen Recall (0.83). Dies ist jedoch primär ein Effekt des strategischen Thresholds, da die zugrundeliegende Trennschärfe mit einer AUC von 0.615 am schwächsten ist. Dies erklärt, warum der hohe Recall nur durch das Inkaufnehmen vieler Falsch-Positiver (FP=266) erreicht werden kann.

Overfitting-Analyse (Train vs. Val)

Die Metriken wurden auf dem Trainings- (N=40.263) und dem Validierungsdatensatz (N=10.066) berechnet, um Overfitting zu prüfen.

Metrik	Training	Validierung	Differenz
Gesamt AUC	0.6905	0.6719	-0.0186
F1-Score (Klasse 1)	0.4156	0.3990	-0.0166
Recall (Klasse 1)	0.6527	0.6315	-0.0212
Precision (Klasse 1)	0.3049	0.2916	-0.0133
Accuracy	0.6271	0.6134	-0.0137

Tabelle 8 XGBoost Modell Overfitting Analyse

Die Performanz ist auf beiden Datensätzen extrem stabil. Die Differenzen bei allen Kernmetriken sind minimal (ca. 1-2 %) (Tabelle 8). Dies deutet darauf hin, dass das Modell nicht überangepasst ist und sehr gut generalisiert.

5.2.2 Regression

Das Ziel der Regressionsmodellierung ist die präzise Vorhersage der zu erwartenden Transaktionsgebühr für jeden einzelnen Zahlungsdienstleister.

Evaluationsmetriken und strategische Ausrichtung

Zur Beurteilung der Vorhersagegüte wurden drei zentrale Kennzahlen herangezogen. Der Mean Absolute Error (MAE) misst die durchschnittliche absolute Abweichung der Prognose von der tatsächlichen Gebühr in Euro und ist somit für den Fachbereich direkt interpretierbar. Ergänzend dazu wurde der Root Mean Squared Error (RMSE) verwendet, der größere Fehler stärker gewichtet und somit essenziell ist, um das Risiko kostspieliger Fehleinschätzungen zu bewerten. Das Bestimmtheitsmaß (R-Quadrat, R^2) diente als unterstützende Metrik, um den Anteil der durch das Modell erklärten Varianz zu quantifizieren. Die strategische Ausrichtung zielte darauf ab, ein Modell zu finden, das einen möglichst niedrigen MAE und RMSE mit einem möglichst hohem R^2 kombiniert.

Baseline-Modell und Auswahl des Datensatzes

Als erster Schritt wurde ein robustes Baseline-Modell mittels linearer Regression etabliert, um einen grundlegenden Leistungs-Benchmark zu definieren. Dieses Modell diente nicht nur als Referenzwert, den jedes komplexere Modell übertreffen muss, sondern wurde auch genutzt, um die Vorhersagekraft der vier unterschiedlichen, im Feature-Selection-Prozess erstellten Datensätze zu vergleichen. Die Experimente zeigten, dass alle vier Datensätze eine identische Performance erzielten. Aufgrund dessen wurde der Datensatz df_Regression_featureSelection für die weitere Modellierung ausgewählt, da er bei gleicher Leistung die geringste Anzahl an Features aufwies. Das Baseline-Modell erreichte auf diesem Datensatz bereits einen vielversprechenden R^2-Wert von 0.697 und einen RMSE von 0.996

auf den Validierungsdaten, was auf eine solide Erklärungskraft und kein Overfitting hindeutete.

Evaluierung und Vergleich anspruchsvollerer Modelle

Nachdem die Baseline etabliert war, wurde eine Palette diverser Regressionsalgorithmen trainiert und evaluiert. Dazu zählten die Ridge-Regression; der SGDRegressor, die PolynomialRegression; sowie zwei leistungsstarke Ensemble-Methoden: der RandomForestRegressor und der LightGBM (LGBM) Regressor. Für jedes Modell wurde mittels Hyperopt eine automatisierte Hyperparameter-Optimierung durchgeführt, um die bestmögliche Leistung zu erzielen. Anschließend wurde im MLflow-Tracking für jeden Algorithmus das Modell mit der Balance aus dem besten MAE, dem besten R^2 und dem besten RMSE identifiziert und als PKL-Datei gespeichert.

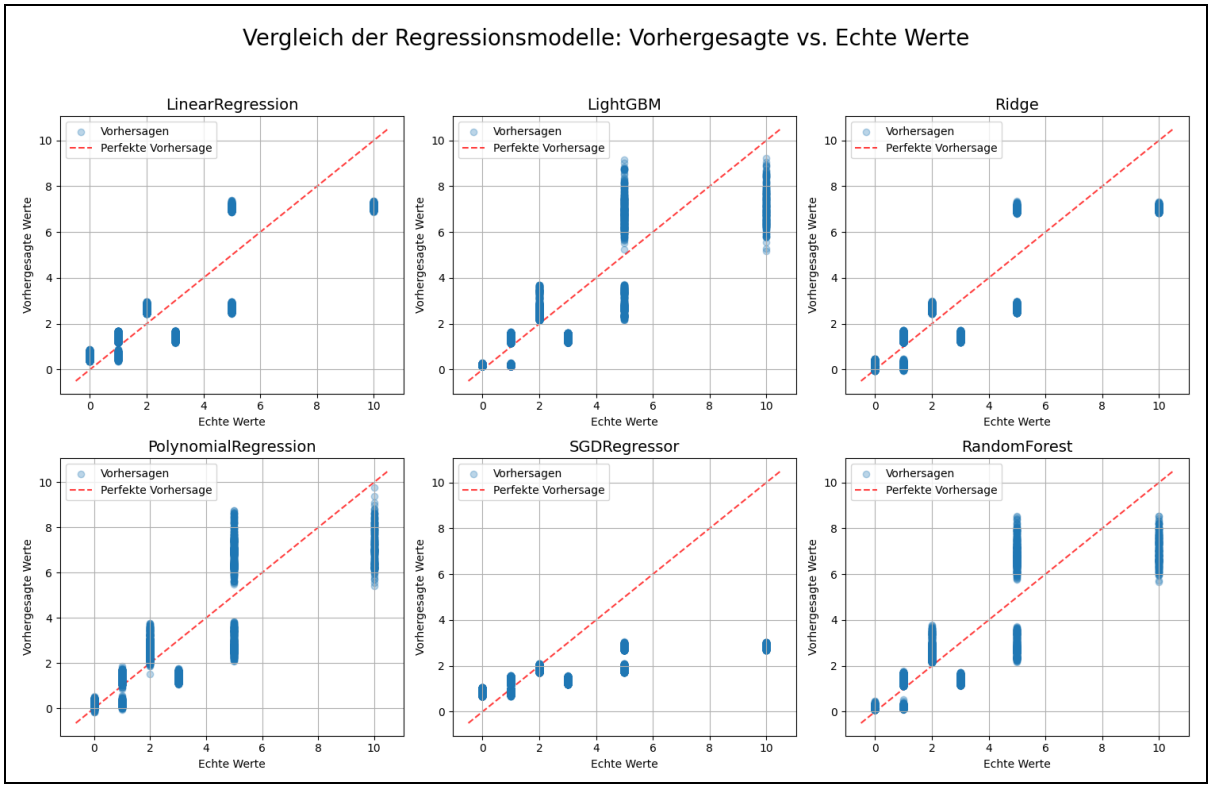

Abbildung 24 Vergleich aller echten vs. Vorhergesagten Werte aller Regressionsmodelle

Die ersten Ergebnisse zeigten eine Überlegenheit der Ensemble-Methoden. Wie im Abbildung 24 visualisiert, lagen die Vorhersagen von LightGBM und RandomForest am engsten an der Linie der perfekten Vorhersage, was auf einen geringeren Fehler hindeutet. Im Gegensatz dazu zeigten die linearen Modelle in kleinen Fee Bereichen sehr gute Vorhersagen. Generell wurden in den höheren Wertebereichen schlechtere Voraussagen getroffen. Dabei fiel vor allem der SGDRegressor negativ auf.

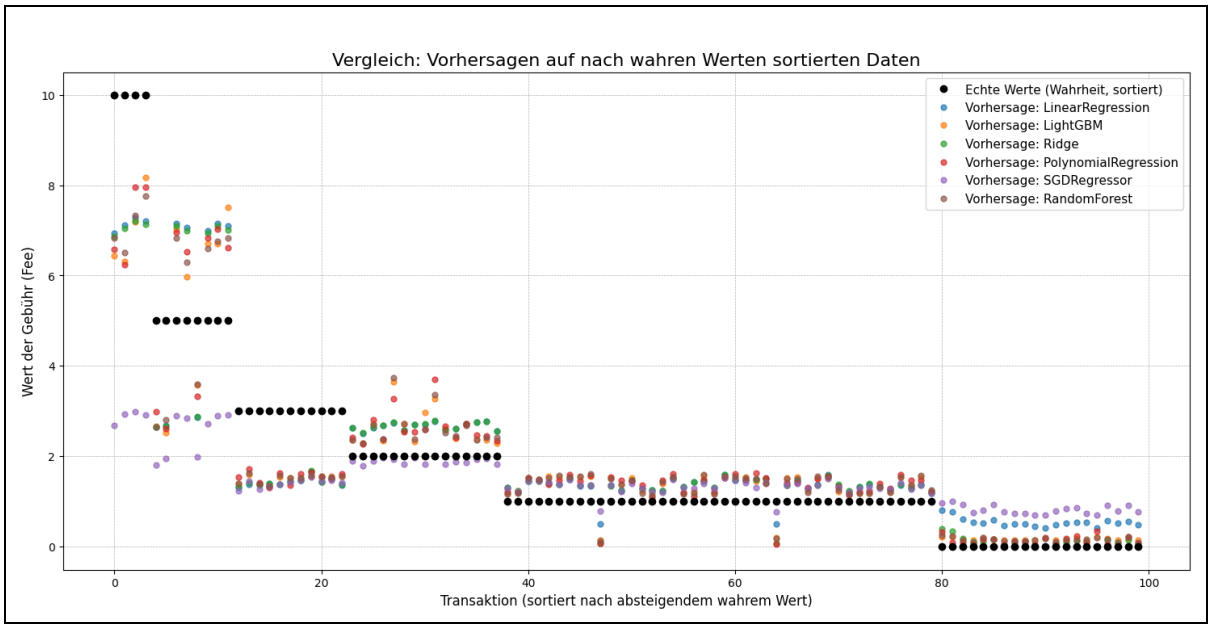

Abbildung 25 Vergleich einer Stichprobe der echten vs. Vorhergesagten Werte aller Regressionsmodelle

Dies ist in Abbildung 25 gut zu erkennen, wo tatsächliche Werte bei höheren Fees weit über den Vorhersagen liegen. Die Ursache hierfür wurde in der ungleichen Verteilung der Daten vermutet, da hohe Gebühren, die oft durch den seltener genutzten PSP Goldcard entstehen, im Datensatz unterrepräsentiert sind.

Gezielte Optimierung durch Gewichtung

Um der systematischen Unterschätzung hoher Werte entgegenzuwirken, wurden gezielte Maßnahmen zur Modellverbesserung eingeleitet. Da die Vermeidung teurer Transaktionen ein strategisches Ziel ist, wurde entschieden, das LightGBM-Modell weiter zu optimieren, indem Stichprobengewichte eingeführt wurden. In einem ersten Schritt wurden Transaktionen mit einem Gebührenwert von 10.0 stärker gewichtet und die Gewichtungsfaktoren selbst in die Hyperparameter-Optimierung einbezogen (LightGBM_value10), in einem zweiten Schritt wurden zusätzlich die Werte von 8.0 gewichtet (LightGBM_value10_8). Der Vergleich dieser gewichteten Modelle mit der ursprünglichen LightGBM-Basislinie zeigte einen klaren Zielkonflikt.

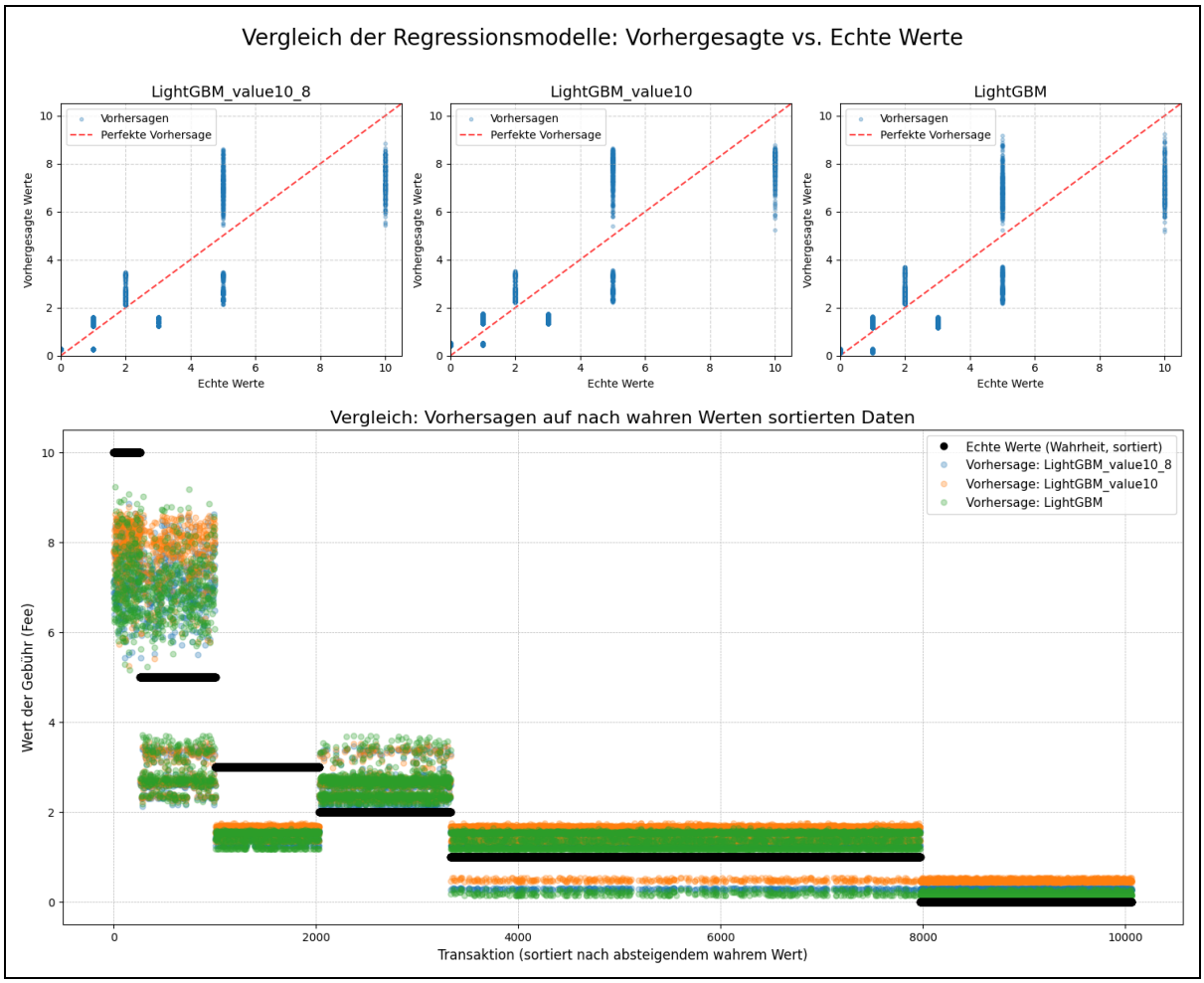

Abbildung 26 Vergleich der echten vs. Vorhergesagten Werte aller LIGHTGBM Modelle

Die Gewichtung verschob die Vorhersagen für die hohen Werte zwar erfolgreich nach oben und reduzierte die Unterschätzung, führte aber gleichzeitig zu einer größeren Ungenauigkeit bei den häufigeren, niedrigen Gebührenwerten (Abbildung 26). Eine Analyse des RMSE pro Gebührenwert bestätigte, dass das Modell LightGBM_value10 den besten Kompromiss darstellte: es reduzierte den RMSE für den kritischen Wert 10 von 2.977 auf 2.246, ohne die mittleren Werte übermäßig zu beeinträchtigen. Aus diesem strategischen Grund – der Priorisierung der korrekten Vorhersage potenziell teurer Transaktionen – wurde das Modell LightGBM_value10 als finales Regressionsmodell ausgewählt.

Finale Analyse des ausgewählten Modells

Das finale, gewichtete LightGBM-Modell stellt ein erfolgreiches Ergebnis der Regressionsanalyse dar. Es erreicht auf den Validierungsdaten einen R^2-Wert von 0.714. Der MAE liegt bei 0,792 €, und der RMSE bei 1,015 €, was auf einen robusten, typischen Vorhersagefehler von etwa einem Euro ohne übermäßige Ausreißer hindeutet. Ein Vergleich der Trainings- und Validierungsmetriken (R^2 von 0.734 vs. 0.714) bestätigt zudem eine exzellente Stabilität und Generalisierungsfähigkeit ohne Overfitting. Die optimierten

Hyperparameter, wie eine sehr niedrige Lernrate (learning_rate = 0.022) und eine relativ große Baumstruktur (max_depth = 15, num_leaves = 75), deuten auf einen tiefen und präzisen Lernprozess hin.

5.2.3 Kombiniertes Modell

Um eine datengestützte und geschäftszielorientierte Empfehlung für den optimalen Zahlungsdienstleister abzugeben, wurde ein kombiniertes Modellsystem entwickelt, das die Stärken eines Klassifikationsmodells zur Vorhersage der Erfolgswahrscheinlichkeit und eines Regressionsmodells zur Schätzung der Transaktionskosten vereint.

Methodischer Ansatz und Modellarchitektur

Das Klassifikationsmodell hat die Aufgabe, für eine gegebene Transaktion die Erfolgswahrscheinlichkeit für jeden der vier PSPs zu prognostizieren. Parallel dazu schätzt das Regressionsmodell die zu erwartenden Transaktionskosten für jeden dieser Anbieter auf einer normalisierten Skala von 0 bis 10. Um eine finale Empfehlung zu generieren, werden die Ergebnisse beider Modelle in einem gewichteten Score zusammengeführt. Die Erfolgswahrscheinlichkeit des Klassifikationsmodells fließt mit einer Gewichtung von 70 % in den Gesamtscore ein, während die prognostizierten Kosten des Regressionsmodells mit 30 % gewichtet und subtrahiert werden, da niedrigere Kosten vorteilhafter sind (Formel 1).

Pipeline der Vorhersage

Der Ablauf von einer Transaktionsanfrage bis zur finalen Empfehlung folgt einer klar definierten Pipeline (Abbildung 27).

```
raw_path = Path(r"./Data/PSP_Jan_Feb_2019.xlsx")
df = pd.read_excel(raw_path)

# === MODELLE LADEN ===
xgb_model = pickle.load(open("XGBoost_auc_enchanting-bee-331.pkl", 'rb'))
lgbm_model_10 = pickle.load(open("lgbm10_crawling-boar-12_d1ad86285a0b4868868f300261aa4364.pkl", 'rb'))

# Scaler außerhalb fitten (einmalig)
scaler = MinMaxScaler(feature_range=(0, 1))
scaler.fit(df[['amount']])

# Amount Bins berechnen (einmalig)
amount_bins = pd.qcut(df['amount'], q=4, labels=[1,2,3,4], retbins=True, duplicates='drop')[1]

# PSP Thresholds definieren (Entscheidungsgrenzen)
PSP_THRESHOLDS = {
 "Simplecard": 0.51,
 "Moneycard": 0.55,
 "Goldcard": 0.60,
 "UK Card": 0.47
}

def predict_psp_scores(amount, secured, card):
 """
 Hauptfunktion: Berechnet Gesamtscore aus Klassifikation (70%) und Regression (30%).
 Verwendet PSP-spezifische Thresholds als Qualitätsfilter mit Fallback.
 """
 # 1. Feature Engineering
 tmsp = datetime.datetime.now()
 hour = tmsp.hour
 monatsanfang = 1 if tmsp.day <= 3 or tmsp.day >= 29 else 0

 daytime_nachmittag = 1 if 13 <= hour < 19 else 0
 daytime_nacht = 1 if 0 <= hour < 7 else 0
```

```
hour_sin = np.sin(2 np.pi hour/24.0)

oh__card_Diners = 1 if card == 'Diners' else 0
oh__card_Visa = 1 if card == 'Visa' else 0
oh__card_Master = 1 if card == 'Master' else 0

amount_cat = pd.cut([amount], bins=amount_bins, labels=[1,2,3,4])[0]
bin__3D_secured = 1 if secured == 'Ja' else 0
remainder__amount_scaled = scaler.transform(pd.DataFrame([[amount]], columns=['amount']))[0][0]

# === REGRESSION SCORES ===
input_dict_regression = {
'bin__3D_secured': bin__3D_secured,
'bin__daytime_Nachmittag': daytime_nachmittag,
'bin__daytime_Nacht': daytime_nacht,
'monatsanfang': monatsanfang,
'oh__PSP_Goldcard': 0,
'oh__PSP_Moneycard': 0,
'oh__PSP_Simplecard': 0,
'oh__PSP_UK Card': 0,
'oh__card_Diners': oh__card_Diners,
'oh__card_Visa': oh__card_Visa,
'ord__amount_cat': amount_cat,
'remainder__amount_scaled': remainder__amount_scaled,
'remainder__hour_sin': hour_sin
}

psp_names = ['Goldcard', 'Moneycard', 'Simplecard', 'UK Card']
psp_columns = [f'oh__PSP_{name}' for name in psp_names]

X_base_reg = pd.DataFrame([input_dict_regression])
regression_scores = {}

for psp_name, psp_col in zip(psp_names, psp_columns):
X_hypo = X_base_reg.copy()

for col in psp_columns:
 X_hypo[col] = 0

X_hypo[psp_col] = 1
pred_value = lgbm_model_10.predict(X_hypo)[0]

# Skalierung von [0, 10] auf [0, 1]
regression_scores[psp_name] = float(np.clip(pred_value / 10, 0, 1))

# === KLASSIFIKATION WAHRSCHEINLICHKEITEN ===
input_dict_classification = {
'oh__card_Visa': oh__card_Visa,
'oh__PSP_Simplecard': 0,
'oh__card_Master': oh__card_Master,
'oh__card_Diners': oh__card_Diners,
'monatsanfang': monatsanfang,
'bin__daytime_Nachmittag': daytime_nachmittag,
'bin__daytime_Nacht': daytime_nacht,
'ord__amount_cat': amount_cat,
'bin__3D_secured': bin__3D_secured,
'oh__PSP_Moneycard': 0,
'oh__PSP_Goldcard': 0,
'remainder__hour_sin': hour_sin,
'remainder__amount_scaled': remainder__amount_scaled,
'oh__PSP_UK Card': 0
}

X_base_clf = pd.DataFrame([input_dict_classification])
classification_probs = {}
classification_decisions = {}

for psp_name in psp_names:
X_hypo = X_base_clf.copy()

# Alle PSPs auf 0 setzen
for psp in psp_names:
 X_hypo[f'oh__PSP_{psp}'] = 0

# Aktuelle PSP auf 1 setzen
X_hypo[f'oh__PSP_{psp_name}'] = 1

# Vorhersage der Wahrscheinlichkeit
prob = float(xgb_model.predict_proba(X_hypo)[:, 1][0])
classification_probs[psp_name] = prob

# Threshold-Check
classification_decisions[psp_name] = 1 if prob >= PSP_THRESHOLDS[psp_name] else 0

# === GESAMTSCORE BERECHNEN ===
# Prüfe, ob mindestens eine PSP den Threshold überschreitet
```

```
any_qualified = any(classification_decisions.values())

final_scores = {}
for psp_name in psp_names:
if any_qualified:
 # Normalfall: Nur qualifizierte PSPs bekommen Score
 if classification_decisions[psp_name] == 1:
 final_scores[psp_name] = (
  classification_probs[psp_name] 0.7 -
  regression_scores[psp_name] 0.3
 )
 else:
 final_scores[psp_name] = None
else:
 # Fallback: Keine PSP qualifiziert → Berechne für ALLE
 final_scores[psp_name] = (
 classification_probs[psp_name] 0.7 -
 regression_scores[psp_name] 0.3
 )

# === BESTE PSP ERMITTELN ===
valid_scores = {psp: score for psp, score in final_scores.items() if score is not None}

if valid_scores:
best_psp = max(valid_scores, key=valid_scores.get)
best_score = valid_scores[best_psp]
fallback_used = not any_qualified
else:
# Sollte nie passieren
best_psp = "Fehler"
best_score = None
fallback_used = False

# === ERGEBNIS MIT ALLEN DETAILS ===
result = {
"Klassifikation_Wahrscheinlichkeiten": classification_probs,
"Klassifikation_Entscheidungen (0=abgelehnt, 1=qualifiziert)": classification_decisions,
"Regression_Scores": regression_scores,
"Gesamtscores": final_scores,
"Empfohlene_PSP": best_psp,
"Bester_Score": round(best_score, 4) if best_score is not None else "Fehler",
"Fallback_Modus": fallback_used,
"Info": "Fallback aktiv: Keine PSP erreicht Threshold" if fallback_used else "Normale Empfehlung: Mindestens eine PSP qualifiziert",
"Angewendete_Thresholds": PSP_THRESHOLDS
}

return result
```

Abbildung 27 Pipeline des Kombinierten Modells als Python Code

Aus den initialen Transaktionsdaten (Betrag, 3D-Secure-Status, Kartentyp) werden in Echtzeit die für die jeweiligen Modelle relevanten Features generiert. Dazu gehören zeitbasierte Merkmale wie die sinus-kodierte Stunde und binäre Indikatoren für "Nacht" und "Nachmittag", der One-Hot-kodierte Kartentyp sowie der kategorisierte und skalierte Betrag. Das System simuliert die Transaktion für jeden der vier PSPs, indem es die entsprechenden One-Hot-kodierten PSP-Merkmale anpasst und die Daten an beide Modelle übergibt. So werden für jeden PSP eine Erfolgswahrscheinlichkeit und ein Kostenscore berechnet. Ein zentrales Element der Logik ist die Anwendung der zuvor ermittelten, PSP-spezifischen Schwellenwerte. Die Erfolgswahrscheinlichkeit jedes PSPs wird mit seinem individuellen Threshold verglichen. Nur PSPs, die diesen Wert erreichen oder überschreiten, gelten als "qualifiziert". Basierend auf der Filterung wird die beste PSP ermittelt. Im Normalfall, wenn mindestens eine PSP qualifiziert ist, wird der Gesamtscore nur für diese berechnet, und der mit dem höchsten Score wird empfohlen. Sollte jedoch kein PSP ihren spezifischen Threshold erreichen, greift ein Fallback-Mechanismus: In diesem Fall werden die Gesamtscores für alle

vier PSPs berechnet, und die beste verfügbare Option wird empfohlen, um sicherzustellen, dass immer eine Handlungsempfehlung gegeben wird.

6 Bereitstellung

Für die praktische Anwendung und Demonstration wurde das gesamte System in eine interaktive Weboberfläche mithilfe der Python-Bibliothek Gradio implementiert (Gradio, n.d.). Es wurden zwei Versionen des Interfaces erstellt: eine detaillierte Version für Analysezwecke, die alle Zwischenergebnisse wie Klassifikationswahrscheinlichkeiten, Regressionsscores, Threshold-Entscheidungen und den finalen Gesamtscore anzeigt, und eine minimalistische Version für den Produktiveinsatz, die lediglich den Namen der empfohlenen PSP ausgibt.

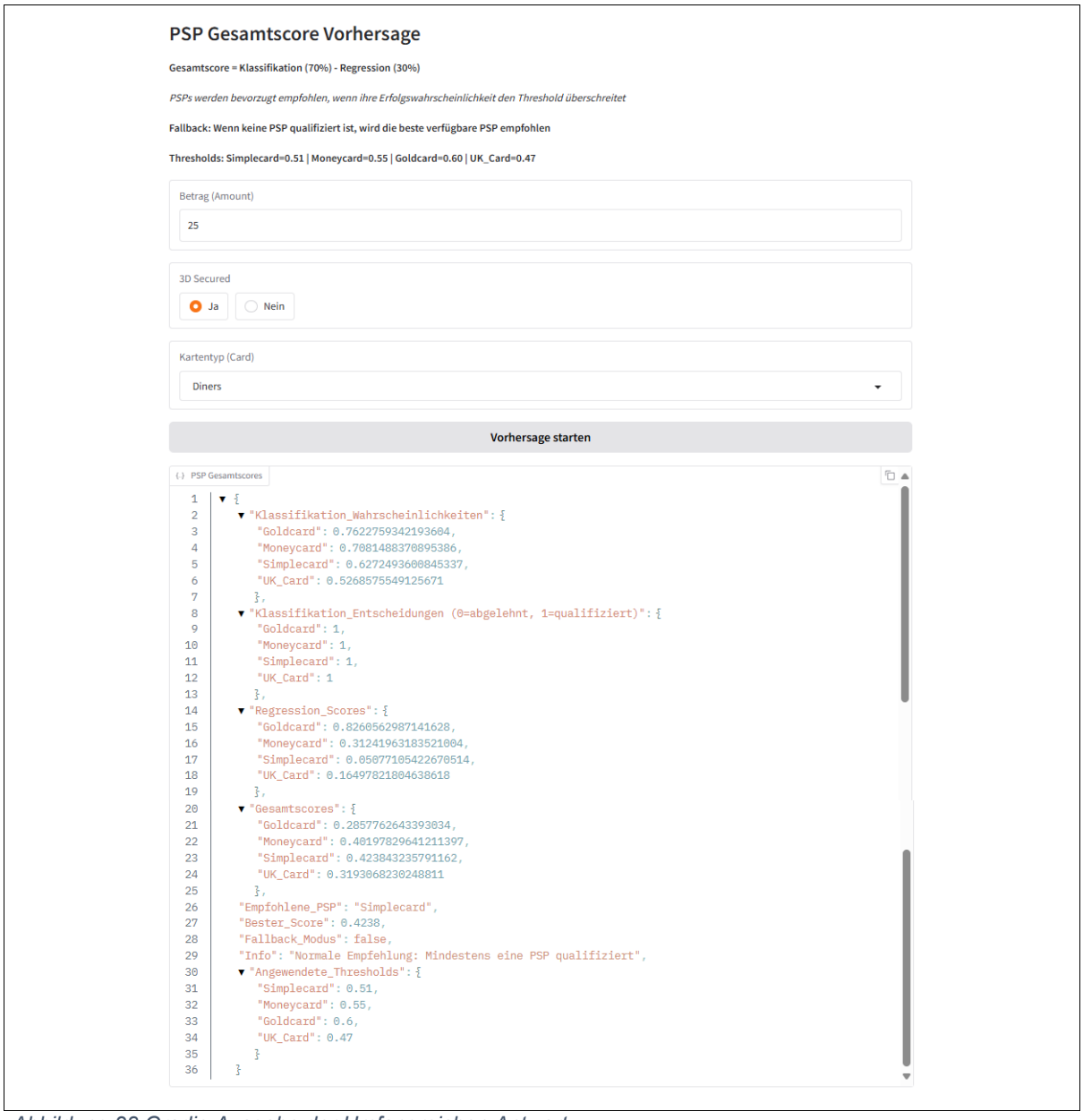

Abbildung 28 Gradio Ausgabe der Umfangreichen Antwort

In Abbildung 28 kann betrachtet werden, wie die Eingabe zu der ausführlichen Ausgabe mit allen Details aussieht. Hier wird trotz der größten Erfolgswahrscheinlichkeit von GoldCard, aufgrund der hohen vorhergesagten Kosten ein anderer PSP gewählt.

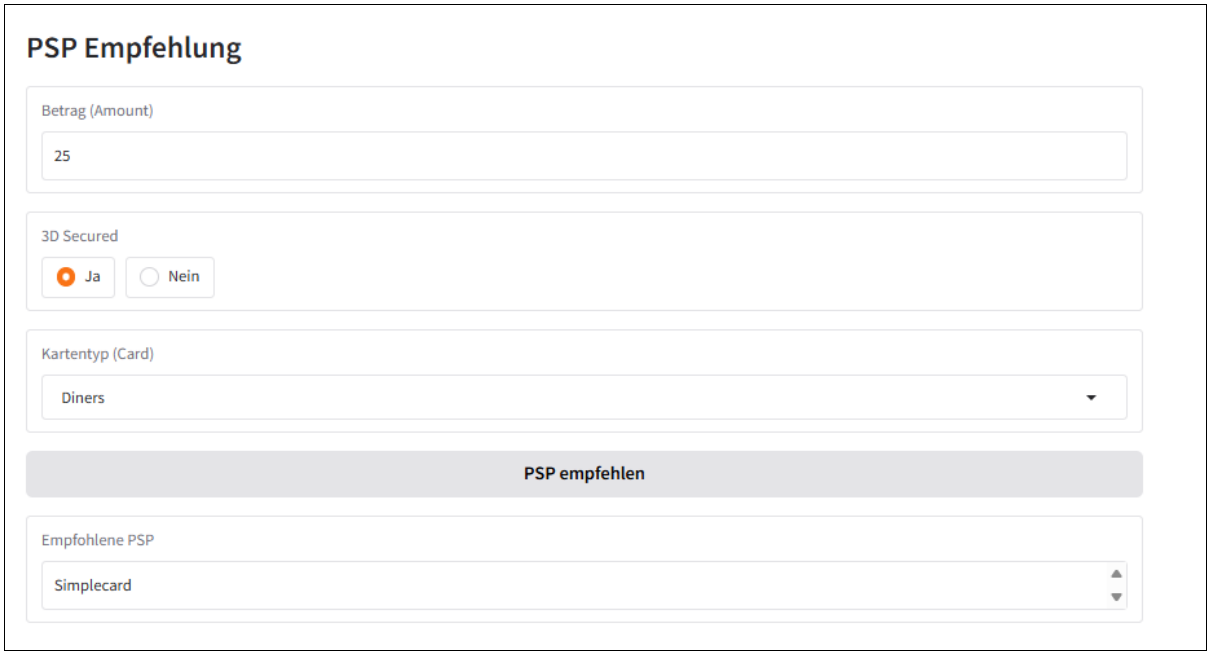

Abbildung 29 Gradio Ausgabe der minimalen Antwort

In Abbildung 29 kann das minimalistische Interface betrachtet werden. Der Nutzer gibt die Transaktionsdetails (Betrag, 3D-Secure-Status, Kartentyp) in die Benutzeroberfläche ein, und nach einem Klick auf den Button wird die Vorhersage-Pipeline durchlaufen und das Ergebnis in Echtzeit angezeigt. Diese Implementierung dient als effektiver Prototyp, der die Funktionalität und den Mehrwert des kombinierten Modellansatzes für ein intelligentes PSP-Routing-System demonstriert. Für den praktischen Einsatz würden bei Kunden automatisch der beste PSP errechnet und angewendet werden.

7 Modell Evaluation

Für Analysezwecke könnte ein Dashboard die Gesamt-Erfolgsrate und die durchschnittlichen Transaktionskosten sowie deren Abweichungen von den Vorhersagen visualisieren, während integrierte Mechanismen zur Erkennung von Daten-Drift bei signifikanten Abweichungen der Eingabedatenverteilungen alarmieren und auf eine eventuell notwendige Neutrainierung des Modells hinweisen.

8 Fazit

Eine der größten Herausforderungen in der Idee, dass ein Machine Learning Modell die Ergebnisse stark verbessert, liegt in der inhärent hohen Fehlerrate der angebundenen Zahlungsdienstleister. Selbst der qualitativ beste Anbieter, Goldcard, weist eine circa 40%ige

Misserfolgsquote auf, während andere, wie Simplecard, in über 80 % der Fälle scheitern. Ein Machine-Learning-Modell kann zwar die bestmögliche Wahl unter den gegebenen Optionen treffen, es kann jedoch die zugrundeliegenden technischen oder prozessualen Mängel der PSPs nicht beheben. Die größte Hebelwirkung zur Steigerung der Gesamterfolgsrate liegt daher nicht allein in der weiteren Modelloptimierung, sondern in strategischen Geschäftsentscheidungen: basierend auf den Ergebnissen muss die Verwendung von Simplecard als Zahlungsanbieter kritisch hinterfragt und potenziell eingestellt werden. Gleichzeitig sollte das 3D-Secure-Verfahren verpflichtend eingesetzt werden, da es sich als einflussreiches Merkmal erwiesen hat.

Des Weiteren offenbarte die Analyse, dass kein einzelnes Merkmal eine überragend starke prädiktive Kraft für den Transaktionserfolg besitzt. Die linearen Korrelationen waren durchweg gering, und prädiktive Muster entstanden erst durch die Kombination mehrerer Merkmale wie PSP-Typ, Betragshöhe, 3D-Secure-Status und spezifische Tageszeiten. Diese Komplexität, gepaart mit der starken Klassen-Imbalance von nur etwa 27 % erfolgreichen Transaktionen, stellt eine natürliche Grenze für die Vorhersagegenauigkeit dar. Für die Zukunft ergeben sich daraus mehrere konkrete Handlungsempfehlungen. Zunächst sollte die Datenerfassung über einen längeren Zeitraum ausgedehnt werden, um saisonale Effekte, sowie weitere Features (. z.B. Systemupdates, Websitebesucher, Serverauslastung) zu erfassen. Zudem könnte eine Änderung der TransaktionsId Logik, etwa durch die Rundung der Zeitstempel (tmsp) auf 60-Sekunden-Intervalle statt auf die volle Minute eine Neubetrachtung der ermöglichen.

Basierend auf der Erkenntnis, dass die Erfolgswahrscheinlichkeit mit jedem Versuch sinkt und die Kosten steigen, könnte eine kostenoptimierende Geschäftsregel implementiert werden, die Transaktionen nach einer bestimmten Anzahl von Fehlschlägen (z. B. drei) automatisch abbricht. Um die Vorhersagegüte weiter zu steigern, könnten die Entwicklung spezialisierter Modelle für jeden einzelnen PSP in Betracht gezogen werden, oder ein stärkeres Augenmerk auf kritische Wochentage (wie Dienstag) gelegt werden. Zudem muss die Balance der Modellkomplexität gewahrt werden. Da aufwändige, kombinierte Modelle rechenintensiver sind, gilt es kritisch abzuwägen, ob dieser Mehraufwand den Nutzen rechtfertigt oder ob schlankere Ansätze bereits ein vergleichbares Ergebnis erzielen.

Abschließend ist es essenziell, das System für den Praxiseinsatz zu überführen. Dies sollte idealerweise über einen API-Call erfolgen, der ein JSON-Objekt zurückgibt, welches flexibel in bestehende Benutzeroberflächen und Backend-Strukturen (z. B. in C#) integriert werden kann. Nur so kann eine robuste, automatisierte MLOps-Pipeline ein kontinuierliches Monitoring der Modellperformance und regelmäßige Neutrainings ermöglichen, um eine nachhaltige Wertschöpfung für das Unternehmen sicherzustellen.

Literaturverzeichnis

Clara Simon. (25. Oktober 2025). *Git Repository dieses Projekts:* . Von https://github.com/DataEnthusiast95/Model_Engineering.git abgerufen

geeksforgeeks. (23. Juli 2025). *www.geeksforgeeks.com*. Von https://www.geeksforgeeks.org/machine-learning/recursive-feature-elimination/ abgerufen

Gradio. (n.d.). *www.Gradio.com*. Abgerufen am 19. Oktober 2025 von https://www.gradio.app/

IBM. (17. August 2021). *www.ibm.com*. Von https://www.ibm.com/docs/de/spss-modeler/saas?topic=dm-crisp-help-overview abgerufen

Jaiswal, S. (16. Janaur 2025). *www.datacamp.com*. Von https://www.datacamp.com/de/tutorial/normalization-in-machine-learning abgerufen

Microsoft. (n.d.). *www.Github.com*. Abgerufen am 18. Oktober 2025 von https://microsoft.github.io/azureml-ops-accelerator/1-MLOpsFoundation/2-SkillsRolesAndResponsibilities/1-AdoptingDSProcess.html

MLFlow. (n.d.). *www.mlflow.com*. Abgerufen am 19. Oktober 2025 von https://mlflow.org/docs/latest/genai/tracing/observe-with-traces/ui

Soetewey, A. (05. September 2023). *www.statsandr.com*. Von https://statsandr.com/blog/pearson-spearman-kendall-correlation-by-hand/ abgerufen

Thakur, K. (01. April 2025). *https://code-b.dev/*. Von https://code-b.dev/blog/encoders-machine-learning abgerufen

Valença, M. (20 August). *www.medium.com*. Von 2024: https://medium.com/@valencamatheus97/sine-and-cosine-transformation-and-normalization-491a6f71c091 abgerufen

Werk 1: scikit-learn. (n.d.). *www.scikit-learn.com*. Abgerufen am 19. Oktober 2025 von https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

Werk 2: scikit-learn. (n.d.). *www./scikit-learn.com*. Abgerufen am 19. Oktober 2025 von https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html