ANALYSE

Verlustprävention an Selbstbedienungskassen im Einzelhandel

Projektgruppe: Raphael Schaffarczik, David Zurschmitten, Matthias Bald
Auftraggeber: Wertkauf GmbH

Abgabedatum: 22.06.2025

Dokumentation – Meilenstein 3: Analyse

Vorbemerkungen

Der dritte Meilenstein im Rahmen des DASC-PM umfasst die **Identifikation geeigneter Analyseverfahren** sowie deren **Anwendung** unter Berücksichtigung einer betriebswirtschaftlich sinnvollen **Bewertungsfunktion** (Kostenfunktion). Die Analyseergebnisse werden auf Basis eines Testdatensatzes evaluiert.

Ziel dieses Meilensteins ist es außerdem, der Wertkauf GmbH konkrete Handlungsempfehlungen aufzuzeigen, um den durch Betrugsfälle verursachten Werteverlust an den Selbstbedienungskassen zu minimieren – bei gleichzeitiger Reduktion unnötiger Kontrollen.

Im dritten Meilenstein nach DASC-PM v1.1 erfolgt die vertiefende Analyse der zuvor bereinigten, aggregierten (Zusammenfassung aller Artikel und Merkmale eines Einkaufs zu einer einzigen Transaktionszeile) und explorativ untersuchten Daten. Ziel ist es, ein praxistaugliches Klassifikations- und Regressionsmodell zu entwickeln, das in Kombination mit einer kostenbasierten Entscheidungslogik eine betriebswirtschaftlich sinnvolle Kontrollstrategie an Selbstbedienungskassen ermöglicht.

Zielstellung des Analyse-Meilensteins

Der Meilenstein „Analyse“ dient der Entwicklung und Bewertung geeigneter analytischer Verfahren zur Erreichung der Projektziele. Im Rahmen dieses Meilensteins werden sechs zentrale Ziele verfolgt:

1. Ableitung **statischer Regeln** zur sofortigen Klassifikation offensichtlicher Betrugsfälle
2. Entwicklung eines **Klassifikationsmodells** zur Vorhersage der Betrugswahrscheinlichkeit
3. Entwicklung eines **Regressionsmodells** zur Schätzung der potenziellen Schadenshöhe bei Betrug
4. **Integration** beider Modelle in eine ökonomisch fundierte Entscheidungsfunktion
5. Durchführung **einer Schwellenwert- und Sensitivitätsanalyse** zur Optimierung der Kontrollstrategie
6. Ableitung betriebswirtschaftlicher **Handlungsempfehlungen** für den operativen Einsatz

Diese Schritte dienen als Grundlage für die spätere Modellbewertung, Operationalisierung und Implementierung im Kundensystem.

Datengrundlage

Nachdem im vorherigen Meilenstein „**Datenbereitstellung**“ die uns von der Wertkauf GmbH bereitgestellten Daten erfolgreich aufbereitet, aggregiert und analysiert wurden, liegt eine valide und repräsentative Datenbasis mit geeignetem Merkmalsraum für die Modellbildung vor. Die vorliegenden Daten basieren auf rund 150.000 Transaktionen aus einem Echtbetrieb von Selbstbedienungskassen im Einzelhandel. Jeder Datensatz entspricht einem abgeschlossenen Kaufvorgang und enthält neben Artikel- und Zeitinformationen auch Zusatzdaten aus Kamerasystemen sowie manuelle und automatische Rückmeldungen. Die vollständige Liste der verwendeten Merkmale findet sich im folgenden Abschnitt.

Die Modelle werden nicht nur über klassische Metriken wie Precision, Recall, F2-Score und R^2 bewertet, sondern zusätzlich anhand der realen wirtschaftlichen Wirkung. Hierzu verwenden wir die von der Wertkauf GmbH vorgegeben Bewertungsfunktion. Auf Basis der Testdaten wird evaluiert, wie hoch die tatsächlichen Kontrollgewinne und -verluste wären, wenn man dem Modell folgt.

Aufgesetzt wird auf dem aggregierten Datensatz aus den Transaktionen, Transaktionspositionen, Produktstammdaten und Filialinformationen:

Merkmale

In Phase 2 des Projekts wurden insgesamt 52 Merkmale entwickelt. Einige dieser Merkmale sind redundant oder haben sich in der explorativen Datenanalyse bei isolierter Betrachtung als nicht signifikant erwiesen. Die Redundanz wurde jedoch bewusst beibehalten, um in Phase 3 analysieren zu können, welche Merkmals**kombinationen** von den Modellen am effektivsten genutzt werden können.

Ein Beispiel hierfür ist das Merkmal Tageszeit, das in drei unterschiedlichen Repräsentationen vorliegt:

- als kategorisches Merkmal mit vier Zeiträumen (Morgen, Mittag, Nachmittag, Abend),
- als Stunde des Tages (kategorial)
- sowie als ordinales Merkmal.

Merkmale, die in der Einzelbetrachtung keine Relevanz zeigten, können in Kombination mit anderen Variablen dennoch wertvolle Informationen liefern. Deshalb wurde auf eine zu frühe Eliminierung verzichtet.

Zudem zeigt sich, dass verschiedene Modelltypen unterschiedlich mit der Merkmalsanzahl umgehen:

- Lineare Modelle (z. B. logistische Regression) profitieren häufig von einer reduzierten, fokussierten Merkmalsbasis,
- während Ensemble-Methoden wie Boosting-Modelle mit einer größeren Zahl von Merkmalen robust und leistungsfähig arbeiten können.

Ein vorschnelles Eliminieren potenziell relevanter Merkmale hätte die Modellleistung negativ beeinflussen können, insbesondere bei komplexeren Verfahren wie Boosting-Algorithmen, die auch mit umfangreichen Feature-Mengen gut umgehen.

Thematische Gruppierung der für die Modellbildung genutzten Merkmale:

Kategorie	**Merkmale**
Preisabweichungen & Rabatte	calculated_price_difference, has_positive_price_difference
Zeitliche Merkmale der Transaktion	day_of_week, days_since_sco_introduction, hour, hour_categorical, daytime, month, transaction_duration_seconds, mean_time_between_scans, max_time_between_scans, time_to_first_scan, time_from_last_scan_to_end
Kundenfeedback	has_feedback, feedback_categorical, feedback_low, feedback_middle, feedback_high, feedback_top
Kamerabasierte Hinweise auf Fehlverhalten	has_camera_detected_wrong_product, has_camera_detected_wrong_product_high_certainty, has_unscanned
Produktkategorien in der Transaktion	has_alcohol, has_bakery, has_beverages, has_convenience, has_dairy, has_frozen_goods, has_fruits_vegetables, has_fruits_vegetables_pieces, has_household, has_limited_time_offers, has_long_shelf_life, has_missing, has_personal_care, has_snacks, has_tobacco
Produktdetails	max_product_price, popularity_min, popularity_max, has_voided, has_sold_by_weight, has_age_restricted, n_voided, n_sold_by_weight, n_age_restricted
Transaktionsbezogene Angaben	payment_medium, cash_desk, total_amount, n_lines
Standortmerkmale / Filialdaten	store_id, location, urbanization

Anforderungen an Analyseverfahren

Im Rahmen des dritten Meilensteins „Analyse“ gemäß DASC-PM ergeben sich spezifische Anforderungen an die Auswahl und Gestaltung der Analyseverfahren. Diese Anforderungen leiten sich sowohl aus den betriebswirtschaftlichen Zielsetzungen des Projekts als auch aus den technischen und organisatorischen Rahmenbedingungen der Wertkauf GmbH ab.

Die Auswahl geeigneter Analyseverfahren basierte daher nicht ausschließlich auf der Modellgüte im engeren Sinne – also Metriken wie Precision, Recall oder F2-Score –, sondern berücksichtigte auch qualitative Anforderungen, die für den operativen Einsatz von zentraler Bedeutung sind.

Die nachfolgenden Kriterien wurden im Rahmen einer **gewichteten Gesamtbewertung** herangezogen. Sie flossen gleichrangig neben der rein statistischen Modellleistung in die Auswahl und Beurteilung der finalen Modellarchitektur ein.

Merkmal	**Beschreibung**
Verständlichkeit	Die Ergebnisse des Modells sollten nachvollziehbar und visualisierbar sein.
Umsetzbarkeit	Das Modell sollte sich ohne großen technischen und personellen Aufwand beim Kunden einsetzbar sein.
Reproduzierbarkeit	Anhand des verfügbaren Codes und der formatierten Trainingsdaten sollen sich bei jeder Anwendung sehr ähnliche Ergebnisse erzielen lassen. Aufgrund der wahrscheinlichkeitstheoretischen Modellierung lassen sich Ergebnisse allerdings nie exakt reproduzieren.
Skalierbarkeit	Das Modell sollte in allen Filialen des Kunden unabhängig von dessen Größe einsetzbar sein und bei leicht geänderter Filialstruktur (bzw. Kaufsegment und Kundschaft) immer noch solide Ergebnisse erzielen. Bei starker Abweichung zwischen Trainingsdaten und zukünftigem gewünschten Einsatz (in gänzlich abweichenden Filialen) empfiehlt sich ein erneutes Training der Modelle auf einem passenderen Datensatz. Entsprechender Programmcode befindet sich in Github und kann jederzeit zu einem Nachtraining verwendet werden.
Robustheit	Die erhaltenen Ergebnisse sollen sich nicht durch kleinere Schwankungen in den Eingabedaten fundamental ändern. Auch soll bei einem erneuten Training eine ähnliche trainierte Architektur des Modells mit ähnlicher Justierung erhalten bleiben.

Modellbildungsprozess

Die Modellbildung erfolgt in mehreren aufeinanderfolgenden Stufen. Diese mehrstufige Vorgehensweise ermöglicht die Kombination verschiedener Methoden und berücksichtigt unterschiedliche Aspekte des Entscheidungsprozesses. Dabei werden unterschiedliche Modelltypen mit jeweils variierenden Parametern eingesetzt, um deren Eignung zu vergleichen und zu bewerten.

Im Folgenden wird die schrittweise Vorgehensweise detailliert beschrieben.

Nach Auswertung der Ergebnisse der Datenexploration ergibt sich das folgende Bild über die verschiedenen Kategorien der Datensätze des Trainingsdatensatzes:

Aufteilung der gelabelten Datensätze nach Kategorien:

Kategorie	**Anzahl Datensätze**	**NORMAL**	**FRAUD**	**Anteil FRAUD (%)**	**Gesamtschaden (€)**
Unscanned	377	0	377	100,0 %	5.088 €
Fehlerhafte-Rabatte	1.521	0	1.521	100,0 %	11.058 €
Übrige Rabatte	9.562	8.401	1.161	12,15 %	7.960 €
Übrige	136.564	134.968	1.596	1,17 %	11.057 €
Gesamt	**148.024**	**143.369**	**4.655**	**3,15 %**	**35.163 €**

- **Kategorie „Unscanned“:** Hierbei handelt es sich um Transaktionen, in denen Positionen vom Kamerasystem hinzugefügt wurden, weil ein Produkt gesichtet, aber nicht gescannt wurde (has_unscanned = True).
- **Kategorie „Fehlerhafte Rabatte**“: Als Rabatte wurden im Rahmen der explorativen Datenanalyse alle Positionen („lines“) identifiziert, bei denen der rechnerische sales_price nicht mit dem tatsächlichen sales_price übereinstimmt (durch Betätigen einer entsprechenden „Rabatte-Taste“)

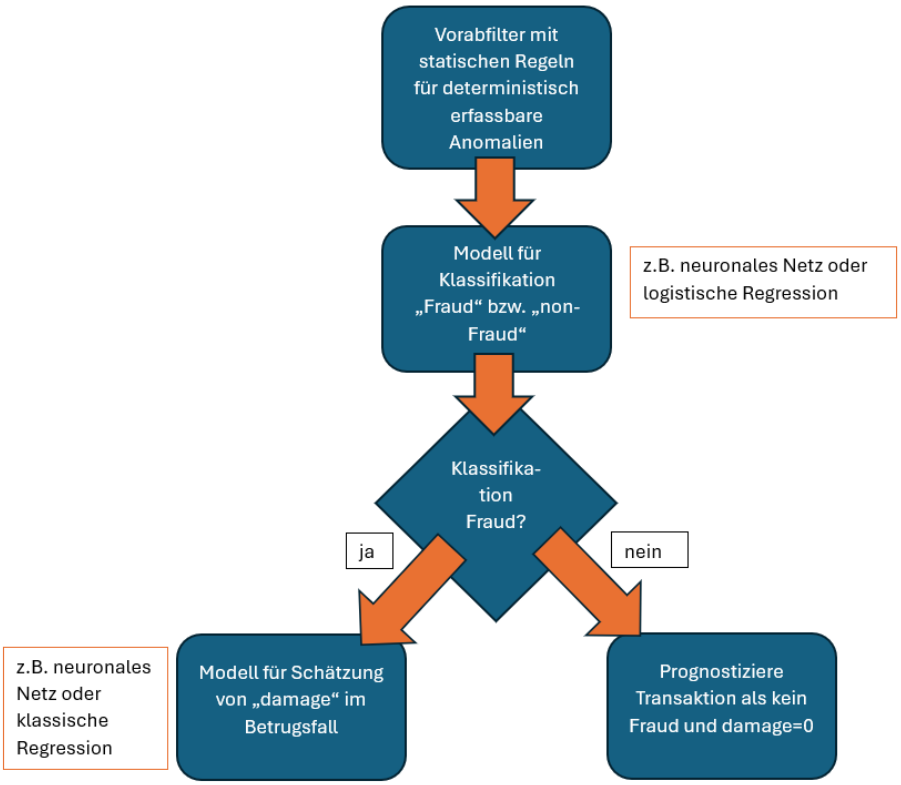
  Transaktionen, in denen für Produktkategorien (z. B. Haushaltswaren) Rabatte aufgrund abgelaufener Mindesthaltbarkeitsdaten (MHD) manuell angewendet wurden, obwohl solche Artikel typischerweise kein MHD aufweisen
- **Kategorie „Übrige Rabatte“:** Diese Kategorie umfasst alle Rabatte, die in Produktkategorien auftreten, in denen reguläre Rabatte im Zusammenhang mit einem abgelaufenen MHD möglich sind. Allerdings ist auch in dieser Kategorie die Wahrscheinlichkeit für einen FRAUD signifikant höher als in den übrigen Datensätzen.
- **Kategorie „Übrige “:** Diese Kategorie umfasst alle übrigen Datensätze, die keiner der oben genannten Kategorien zugeordnet werden können.

Die Rabattsystematik ist auffällig. Wie bereits in Meilenstein 2 diskutiert, bieten sich ggf. statische Regeln in Bezug auf Rabattbetrug an. Hier ist allerdings genau zu differenzieren, welche Filialen welche Produkte konkret mit Rabatten verkaufen und ob Rabattbetrug daher auf konkrete Regeln zurückzuführen ist (z.B. kein Rabatt bei Produkten ohne Mindesthaltbarkeitsdatum). Wir konnten diese Systematik nicht abschließend klären und verzichten im Rahmen der Generalisierbarkeit der Regeln auf zukünftige Filialen auf solche einfachen Rabattregeln. Ein systematischer Zusammenhang geht weiterhin in die Klassifikations- und Regressionsmodelle ein, dort aber in weniger klarer und eindeutiger Form.

Wie bereits in Meilenstein 2 skizziert, bauen wir unser Modell auf drei Säulen auf. Zunächst sind da statische Regeln, die aus prägnanten Attributen einer Transaktion eine einfache Vorabfilterung nach Betrugsfällen vornehmen sollen. Diese sollen besser funktionieren als die Klassifikation per komplexem Modell. Statische Regeln behandeln wir im nächsten Kapitel.

Im Anschluss erläutern wir das Modell zur Klassifikation mittels maschinellen Lernens. Hier sollen komplexe Beziehungen zur Vorhersage von Betrugsfällen modelliert werden, die über einfache statische Regeln hinausgehen. Hier spielt der konkrete Schaden einer Transaktion keine Rolle, lediglich die Vorhersage von Betrugs- bzw. Nicht-Betrugsfällen an sich.

Im dritten Schritt widmen wir uns der konkreten Schadensvorhersage. Da wir in Schritt 2 bereits einen Klassifikator erhalten, der uns Betrugs- bzw. Nicht-Betrugsfälle filtert, können wir uns hier auf die Schätzung von Schäden im Schadensfall (also jenen Fällen, in denen der Klassifikator von Betrug ausgeht) beschränken. Im folgenden Schaubild ist das Gesamtmodell verdeutlicht.



Stufe 1: Statische Regeln zur Vorfilterung

Schritt eins bestand aus der Identifikation einfacher, interpretierbarer Entscheidungsregeln, die sich auf Erkenntnisse der explorativen Datenanalyse stützen und in Meilenstein 3 systematisch untersucht wurden. Ziel war es, Regeln zu finden, die auf ein bis zwei Merkmale (Features) basieren, um eine hohe Verständlichkeit, geringe Komplexität und praktische Anwendbarkeit zu gewährleisten, zugleich eine höhere Präzision bei der Identifikation von Betrugsfällen bieten als der im nächsten Kapitel vorgestellte Klassifikator (auf Basis maschinellen Lernens). Auch der konkrete wirtschaftliche Mehrwert durch Schadensvermeidung wurde bei diesen Regeln ausgewertet.

Methodisches Vorgehen

Die Analyse basierte auf einer binär codierten Version des Datensatzes aggregierten Datensatzes. Die Klassifikation FRAUD/NORMAL wurde als Zielgröße verwendet. Alle kategorialen Merkmale wurden in binäre Indikatoren überführt. Anschließend wurden zwei Analyseschritte durchgeführt:

1. Einzelfeature-Regeln: Für jedes kategoriale Merkmal wurde überprüft, ob ein bestimmter Merkmalsausprägungswert (z. B. has_unscanned == True) eine signifikant erhöhte Trefferquote bei FRAUD-Fällen aufweist.
2. Zweierregeln: Zusätzlich wurden alle möglichen Kombinationen von zwei Merkmalen (Konjunktion: A == True AND B == True) geprüft, sofern sie nicht zu selten auftreten, um nicht nur eine trennscharfe, sondern auch sinnvoll einsetzbare Regel zu erhalten.

Zentrale Attribute bei der Regelbildung waren has_unscanned und has_missing. Es handelt sich um Transaktionen, bei denen von der Kamera fehlende Positionen erkannt wurden und automatisch in die Transaktion als FRAUD aufgenommen wurden (has_unscanned) bzw. Transaktionen mit einer fehlenden Produktidentifikation.

Ergebnisse

Die besten Einzelfeature-Regeln zeigten beeindruckende Präzision, aber teilweise geringe Abdeckung. So konnte beispielsweise die Regel:

- has_unscanned == True
  alle zutreffenden Fälle korrekt als FRAUD erkennen (Precision = 1.0), aber deckte nur etwa 8 % aller FRAUD-Fälle ab (Recall = 0.08). Die False Positive Rate betrug 0.0, was bedeutet, dass kein NORMAL-Fall fälschlich als FRAUD eingestuft wurde. Die Regel hätte insgesamt einen wirtschaftlichen Schaden von über 5.000 € verhindert.

Eine ähnliche Beobachtung wurde für die Regel has_missing == True gemacht, die ebenfalls sehr präzise ist, jedoch nur eine Teilmenge der Betrugsfälle erkennt. Kombinierte Regeln (z. B. has_unscanned == True AND feedback_categorical == "low") konnten die Erkennungsquote erhöhen, allerdings meist zulasten der Präzision.

	Regel	TP	FP	FN	TN	Precision	Recall	FPR	FNR	Verhinderter Schaden
0	has_unscanned == True	377	0	4278	143369	1.0	0.080988	0.0	0.919012	5088.38
1	has_missing == True	16	0	4639	143369	1.0	0.003437	0.0	0.996563	200.07

Bewertung

Die Untersuchung zeigt, dass einfache Regeln bereits sinnvoll zur Risikoabschätzung eingesetzt werden können. Besonders für operative Maßnahmen im Echtzeitbetrieb (Hinzuziehen eines Mitarbeiters) eignen sich diese Regeln als Filter geringer Komplexität. In unserem finalen Modell verwenden wir die beiden Erkennungsfilter has_unscanned bzw. has_missing, eine FPR (Anteil der als Betrug klassifizierten Fälle, die aber kein Betrug sind) von 0 aufweisen. D.h. immer dann, wenn diese Regel anschlägt, kann zumindest gemäß Trainingsdaten von einem Betrugsfall bzw. einer fehlerhaften Transaktion ausgegangen werden.

Die meisten Fälle werden nicht von dieser Regel abgedeckt, da has_unscanned und has_missing nur selten wahr sind. In den Fällen, in denen diese Merkmale anschlagen, kann durch die direkte Klassifikation als FRAUD jedoch Präzision gegenüber dem im folgenden Abschnitt erläuterten Klassifikator gewonnen werden.

Weitere Regeln eigenen sich nicht als statische Vorabregeln, da sie entweder kaum Fälle abdecken und daher einerseits in der Implementierung nicht sinnvoll scheinen, andererseits auch das Risiko von Überanpassung besteht. Zum Beispiel ein Einkauf, der sowohl Snacks hat als auch eine ungewöhnlich lange Transaktionsdauer, eine bestimmte Warenkorbgröße und zu einer gewissen Tageszeit stattfindet, ist wenig geeignet für eine generalisierende Regel.

Stufe 2: Klassifikation der Transaktionen

In diesem Kapitel wird die Entwicklung des Klassifikationsmodells beschrieben – eines zentralen Bestandteils unseres analytischen Ansatzes zur Erkennung potenzieller Fraud-Fälle. Die gängigen Klassifikationsmodelle geben für jede beobachtete Transaktion einen Wert zwischen 0 und 1 aus. Dieser Wert drückt aus, wie stark das Modell eine bestimmte Klasse favorisiert – in unserem Fall, ob es sich um einen Fraud-Fall handelt oder nicht.

Obwohl diese Werte häufig als „Wahrscheinlichkeiten“ interpretiert werden, handelt es sich technisch nicht um echte Wahrscheinlichkeiten im mathematischen Sinne, sondern um modellinterne Scores. Diese Scores können – insbesondere bei gut kalibrierten Modellen – eine sinnvolle Annäherung an Wahrscheinlichkeiten darstellen, sind aber stark von der Modellart, Trainingsdaten und Kalibrierung abhängig.

Ein Wert nahe 0 bedeutet, dass das Modell eine legitime Transaktion erwartet, während ein Wert nahe 1 auf einen vermuteten Fraud-Fall hinweist. Um daraus eine konkrete Entscheidung abzuleiten, wird in der Regel ein Schwellenwert (Threshold) definiert, ab dem eine Transaktion als verdächtig eingestuft wird.

Iterativer Modellentwicklungsprozess

Die Entwicklung des Klassifikationsmodells erfolgte in einem iterativen und selektiven Prozess, mit dem Ziel, ein leistungsfähiges und robustes Modell für die Identifikation potenzieller Fraud-Fälle zu erstellen. Dabei wurden verschiedene Modellvarianten ausprobiert, jedoch nicht alle in gleichem Umfang weiterverfolgt. Da Training, Kalibrierung und Merkmalsauswahl mit erheblichem Aufwand verbunden sind, wurden nur vielversprechende Ansätze vertieft entwickelt.

Der Entwicklungsprozess umfasste folgende Schritte:

1. **Modellauswahl und erste Tests**
   Es wurden mehrere Klassifikationsverfahren evaluiert, die für binäre Entscheidungen auf tabellarischen Daten geeignet sind. Weniger geeignete Verfahren – z. B. solche mit langen Inferenzzeiten oder schwacher Leistung in frühen Tests – wurden frühzeitig ausgeschlossen. Für die vielversprechenderen Modelle wurde der Optimierungsprozess weitergeführt.
2. **Hyperparameteroptimierung und Kalibrierung**
   Bei den weiterverfolgten Modellvarianten wurden die wichtigsten Hyperparameter gezielt angepasst, um ein gutes Verhältnis zwischen Modellkomplexität und Generalisierungsfähigkeit zu erreichen.
3. **Merkmalsauswahl**
   Parallel zur Modelloptimierung wurden Merkmale auf ihre Relevanz geprüft. Ziel war es, irrelevante oder gar kontraproduktive Variablen zu identifizieren und zu entfernen. Dies dient sowohl der Modellvereinfachung als auch der Verbesserung der Robustheit gegenüber neuen Daten.
4. **Modellvergleich und Evaluation**
   Die Modelle wurden anhand geeigneter Metriken systematisch verglichen. Aufgrund der stark unausgeglichenen Klassenverteilung lag der Fokus auf Kennzahlen, die mit dieser Herausforderung robust umgehen können. Neben Precision, Recall wurde insbesondere die „Area Under the Curve“ der Precision-Recall-Kurve (AUC-PR) verwendet, da sie gezielt die Modellleistung auf der relevanten Klasse (Fraud) abbildet. Darüber hinaus floss die von der Wertkauf GmbH vorgegebene Bewertungsfunktion in die Evaluation ein, da sie es ermöglicht, die praktische Güte eines Modells bei gegebenen Precision- und Recall-Werten im spezifischen Anwendungskontext realistisch einzuschätzen.

   Neben der reinen Leistungsfähigkeit wurden auch Aspekte wie Interpretierbarkeit und Rechenaufwand berücksichtigt.

Verwendete Modelle

Als Basismodelle wurden zunächst das in Phase 2 entwickelte lineare Modell (Logistische Regression) sowie ein einfacher Entscheidungsbaum eingesetzt. Diese dienten als Referenzpunkt für die Bewertung komplexerer Verfahren.

Im weiteren Verlauf wurden verschiedene fortgeschrittene Modellklassen getestet, darunter Random Forests, Boosting-Modelle[1] und einfache neuronale Netze. Da insbesondere die Boosting-Modelle gut abschnitten, wurden diese beiden Modellvarianten gezielt weiterentwickelt und verfeinert.

Neuronale Netze zeigten zunächst ebenfalls vielversprechende Ergebnisse, insbesondere bei einzelnen Konfigurationen. In mehreren Experimenten konnten gute Precision- und Recall-Werte erzielt werden – allerdings erwiesen sich die Resultate als stark abhängig von Initialisierungen und Trainingsparametern.

Trotz wiederholter Versuche und verschiedener Architekturvarianten war die Konsistenz der Trainingsverläufe nicht ausreichend gewährleistet.

[1] Ensemble-Modelle kombinieren die Vorhersagen mehrerer schwächerer Modelle („weak learners“), um die Gesamtleistung zu steigern. Typische Vertreter sind Random Forest (Bagging-Ansatz) sowie Boosting-Methoden wie XGBoost und CatBoost, die iterativ trainieren und dabei Fehler der Vorgängermodelle gezielt ausgleichen.

Modellvergleich:

- Datengrundlage: für das Training und Evaluation wurden die gelabelten Daten, ohne die durch die statischen Regeln (has_unscanned, has_missing) klassifizierbaren Transaktionen gewählt
- Für das Lineare Modell wurden nur die 3 in Phase 2 als relevant bestimmten Merkmale verwendet
- Die übrigen Modelle verwenden die 29 folgenden Merkmale (die übrigen Merkmale hatten keinen bzw. einen negativen Einfluss auf die Ergebnisse)
  payment_medium, total_amount, n_lines, has_feedback, feedback_categorical, daytime, hour, month, n_voided, n_age_restricted, n_sold_by_weight, popularity_min, calculated_price_difference, has_beverages, has_personal_care, has_household, has_tobacco, has_fruits_vegetables_pieces, has_convenience, has_long_shelf_life, has_dairy, has_snacks, has_frozen_goods, has_alcohol, mean_time_between_scans, max_time_between_scans, time_from_last_scan_to_end, days_since_sco_introduction, has_camera_detected_wrong_product_high_certainty
- One-Hot-Encoding der kategorialen Variablen und Skalierung (falls nötig)
- Zur Bewertung und zum Vergleich der Modellleistung wurde ein wiederholtes Kreuzvalidierungsverfahren angewendet: 5-fache Kreuzvalidierung unter Beibehaltung der Klassenverteilung (engl. stratification), jeweils 80 % für das Training und zu 20 % für die Validierung innerhalb jedes Folds. Dies bei 5 Wiederholungen, sodass insgesamt 25 unterschiedliche Trainings-/Validierungskombinationen entstanden.

Ergebnis des Modellvergleichs:

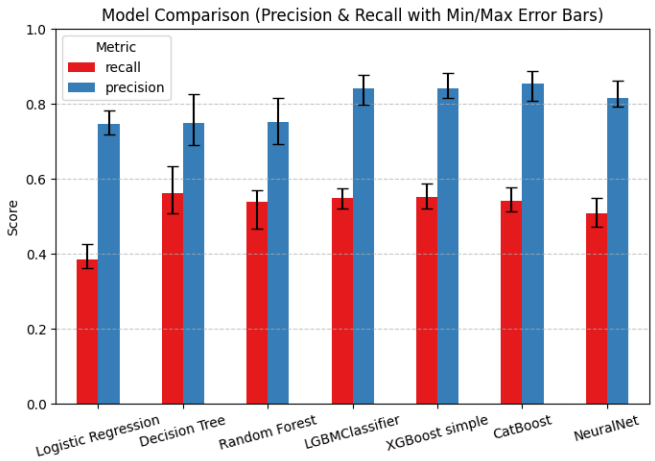
Arithmetische Mittelwerte ausgewählter Metriken über alle Testruns:

	precision	recall	f1	auc-pr	damage_prevented	Bewertung
Logistic Regression	0.746	0.385	0.508	0.431	2219.208	-3271.956
Decision Tree	0.749	0.561	0.641	0.655	3648.474	-1585.290
Random Forest	0.753	0.540	0.628	0.681	3484.541	-1748.223
LGBMClassifier	0.843	0.549	0.664	0.729	3524.914	-1020.850
XGBoost simple	0.842	0.552	0.667	0.730	3555.049	-982.715
CatBoost	0.854	0.543	0.664	0.733	3510.251	-978.913
NeuralNet	0.816	0.508	0.626	0.681	3356.139	-1468.625

Arithmetische Mittelwerte der Konfusionsmatrizen über alle Testruns:

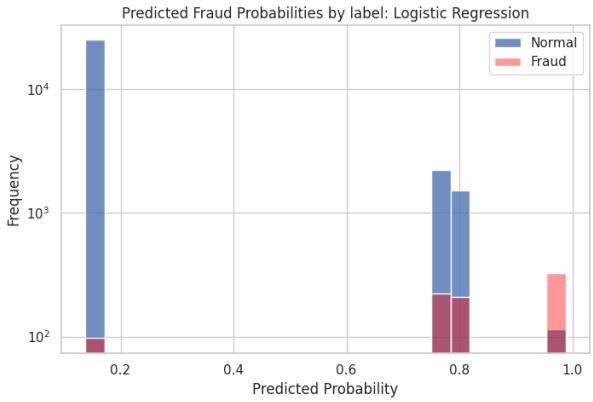
	TP	FP	FN	TN
Logistic Regression	329.20	112.20	526.40	28561.60
Decision Tree	480.28	162.00	375.32	28511.80
Random Forest	462.16	152.84	393.44	28520.96
LGBMClassifier	469.40	87.76	386.20	28586.04
XGBoost simple	472.60	88.56	383.00	28585.24
CatBoost	464.56	79.68	391.04	28594.12
NeuralNet	434.72	98.32	420.88	28575.48

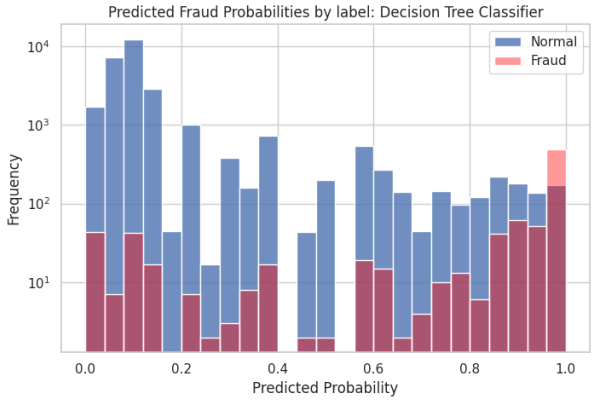
Die Boosting-Modelle (LightGBM, XGBoost, CatBoost) liefern die insgesamt besten Ergebnisse sowohl in der Klassifikationsgüte als auch in der wirtschaftlichen Relevanz. Sie überzeigen mit einer hohen Präzision (wenig unnötige Kontrollen) und erkennen ca. 55% der Fraud-Fälle. Das sind 5% mehr als das neuronale Netz, das ebenfalls eine recht hohe Präzision aufweist. Klassische Modelle wie Logistic Regression sind in diesem Szenario weniger geeignet, da sie vor allem bei der Erkennung von Betrugsfällen (Recall) und dem potenziell verhinderten Schaden deutlich schlechter abschneiden.

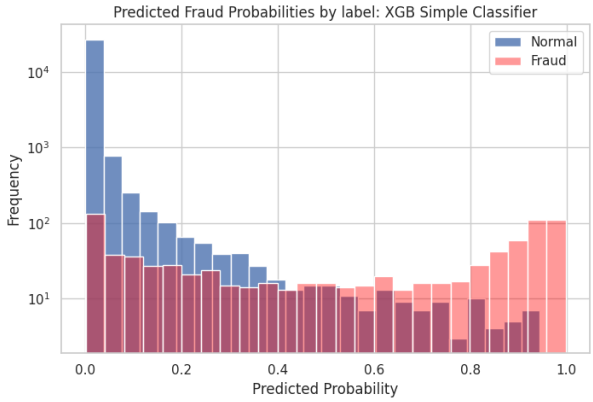


Die Unterschiede in der Performance zwischen den getesteten Boosting-Modellen (insbesondere XGBoost und CatBoost) sind relativ gering. XGBoost hat einen besseren Recall, erkennt also mehr Fraud-Fälle, schneidet aber bei der Präzision schlechter als CatBoost ab und würde unnötige Kontrollen verursachen.

Erstaunlich ist, dass der Random Forest im Vergleich zum einzelnen Entscheidungsbaum in mehreren Metriken – insbesondere Recall und F1-Score – etwas schlechter abschneidet. Üblicherweise erzielt Random Forest durch das Aggregieren vieler Bäume eine stabilere und oft bessere Performance. Ein möglicher Grund für dieses Ergebnis könnte sein, dass bei beiden Modellen keine umfangreiche Hyperparameter-Optimierung durchgeführt wurde. Ohne gezieltes Finetuning kann Random Forest seine Vorteile nicht voll ausspielen und bleibt möglicherweise hinter einem gut konfigurierten Einzelbaum zurück.







Entscheidung für XGBoost:

Obwohl CatBoost im Test geringfügig besser abschnitt, haben wir uns für das XGBoost-Modell entschieden. Ausschlaggebend für diese Entscheidung waren praktische Erwägungen: XGBoost ist weiter verbreitet, besser dokumentiert und im Training performanter, was auch für die spätere Wartung und Weiterentwicklung des Systems von Vorteil ist. In Summe überwiegen damit die praktischen Vorteile von XGBoost gegenüber den minimalen Performance-Unterschieden.

XGBoost erfüllt außerdem alle oben definierten Anforderungen an die Analyseverfahren:

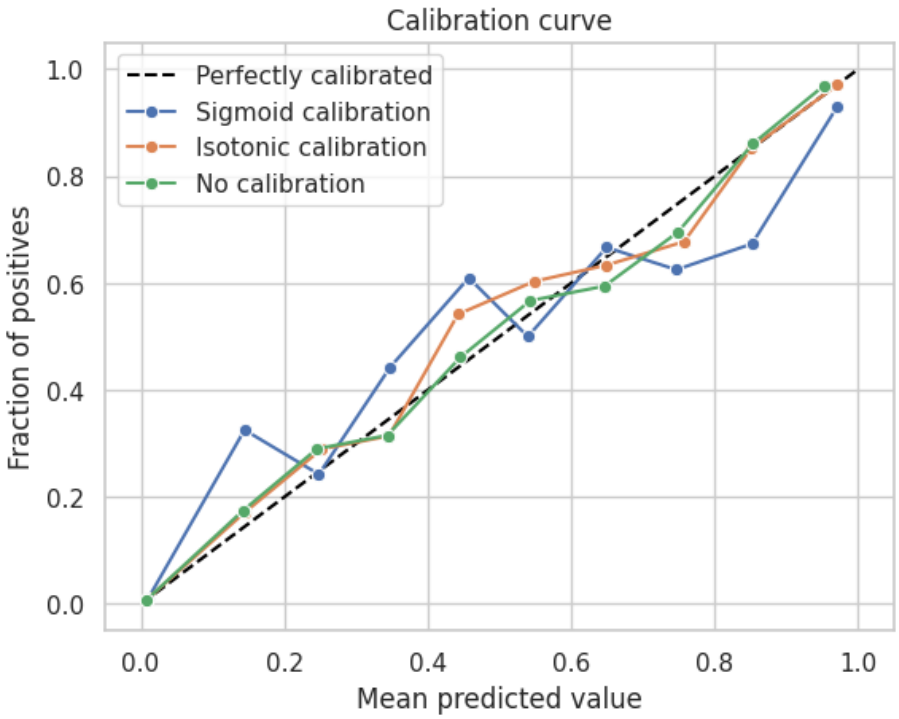
1. **Verständlichkeit (Erklärbarkeit)**:
   Trotz seiner Komplexität als Ensemble-Verfahren auf Basis von Entscheidungsbäumen bietet XGBoost eine gute Interpretierbarkeit der Vorhersagen. Durch die Verwendung etablierter Methoden wie Feature Importance, SHAP-Werte oder Partial Dependence Plots können die Einflussgrößen einzelner Merkmale nachvollziehbar dargestellt und analysiert werden. Dies ermöglicht eine transparente Kommunikation der Modellentscheidungen gegenüber Fachbereichen und Management.
2. **Umsetzbarkeit**:
   XGBoost ist als etablierte Open-Source-Bibliothek breit verfügbar und wird von vielen produktiven ML-Plattformen unterstützt. Es lässt sich effizient in bestehende Software-Infrastrukturen integrieren und erlaubt sowohl Inferenz in Echtzeit als auch die Verarbeitung großer Datenmengen. Darüber hinaus bietet XGBoost eine Vielzahl an Parametern zur Feinjustierung, was eine Anpassung an kundenspezifische Anforderungen erlaubt – etwa bei der Priorisierung von False Positives oder Laufzeitrestriktionen.

3. **Reproduzierbarkeit**:
   XGBoost ermöglicht eine konsistente Reproduzierbarkeit der Ergebnisse durch kontrollierte Zufallssaaten, stabile Modellarchitektur und deterministische Trainingsprozesse (sofern gewünscht). Das gesamte Trainings- und Evaluationsverfahren lässt sich dokumentieren und versionieren, was sowohl intern als auch extern für Transparenz und Nachvollziehbarkeit sorgt.
4. **Skalierbarkeit**:
   XGBoost ist für große Datenmengen optimiert. Es unterstützt verteiltes Training, Sparse-Matrix-Verarbeitung sowie GPU-Beschleunigung. Damit ist es auch bei wachsenden Datenbeständen performant einsetzbar, ohne dass ein Wechsel des Modells erforderlich wird.
5. **Robustheit**:
   XGBoost geht effektiv mit verrauschten Daten, fehlenden Werten und multikollinearen Merkmalen um. Es ist wenig anfällig gegenüber Ausreißern und Overfitting, dank eingebauter Regularisierung. Dies macht es besonders geeignet für reale, heterogene Datenumgebungen.

Schwellwertoptimierung und Kalibrierung des XGBoost-Klassifikators:

Wie bereits angedeutet, können die Score Werte des Klassifikationsmodells prinzipiell nicht direkt als Wahrscheinlichkeiten interpretiert werden, sofern sie nicht ordnungsgemäß kalibriert wurden. Manche Modelle kalibrieren den Output von vornherein gut (z.B. klassische logistische Regression). Andere wiederum, wie z.B. der XGBoost Algorithmus ist in der Regel schlecht kalibriert. D.h. ein klassischer Schwellwert von 0,5 als Übergang von einer Klassifizierung als NORMAL zu einer als FRAUD ist nicht korrekt. Hier muss der Wert an die vorhandene Skala der Output-Werte angepasst werden.

Wie der folgende Plot der Kalibrierungskurve zeigt, ist das Modell bereits gut kalibriert. Eine nachträgliche Rekalibrierung mittels zweier verschiedener Regressionsmethoden (Sigmoid- und isotonische Regression) führt hingegen zu einer Verschlechterung: Die resultierenden Kurven entfernen sich weiter von der Idealgeraden.



Die nachträglich kalibrierten Klassifikatoren zeigten in den relevanten Downstream-Tasks - insbesondere bei der Vorhersage der Bewertungsfunktion im Rahmen des kombinierten Modells aus Klassifikation, Schadensschätzung und Entscheidungsregel und der mittels Bewertungsfunktion bewerteten FRAUD-Vorhersage - durchweg schlechtere Ergebnisse als das ursprüngliche, unkalibrierte Modell. Dieses Ergebnis deutet darauf hin, dass das Ausgangsmodell bereits eine hinreichend gute Kalibrierung aufweist und die ausgegebenen Scores weitgehend als probabilistische Einschätzungen interpretiert werden können. Eine nachträgliche Kalibrierung mit den verwendeten Methoden ist kontraproduktiv.

Eine isolierte Schwellwertoptimierung anhand der Trainingsdaten ergab einen optimalen Schwellenwert bei ca. 0.42, was auf den ersten Blick gegen eine ideale Kalibrierung des Klassifikationsmodells spricht. Bei perfekt kalibrierten Modellen würde man einen optimalen Schwellwert nahe 0.5 erwarten. Allerdings zeigt sich bei der Anwendung dieses Schwellwerts auf unabhängigen Testdaten, dass die Performance tendenziell schlechter ausfällt als beim ursprünglichen Wert (0.5). Auch andere Schwellwerte im Bereich zwischen 0.4 und 0.5 liefern keine stabilen Verbesserungen.

Ein möglicher Erklärungsansatz hierfür ist, dass der scheinbare "Optimal-Schwellwert" auf die Trainingsdaten überangepasst ist. Hinzu kommt noch das stark unausgeglichene Klassenverhältnis. Solche lokal optimierten Schwellenwerte tendieren dazu, die Modellentscheidung stärker in Richtung der grösseren Klasse zu verschieben, was im Validierungskontext – bei leicht anderen Verteilungen oder Decision-Trade-Offs – zu einem schlechteren Verhältnis von Precision und Recall führen kann.

Das robuste Abschneiden des Modells mit einem Schwellwert von 0.5 spricht dafür, dass die Scores des Modells bereits gut kalibriert sind. Ein zusätzliches Finetuning des Schwellenwerts bringt keinen nachhaltigen Nutzen und birgt sogar das Risiko, die Generalisierungsfähigkeit des Modells zu verschlechtern.

Stufe 3: Regressionsmodell zur Schadensschätzung

Ziel dieser Modellkomponente ist es, die finanzielle Schadenshöhe zu prognostizieren, die im Falle eines nicht erkannten Betrugs entsteht. Dabei wird unabhängig davon, wie wahrscheinlich eine Transaktion tatsächlich als betrügerisch eingestuft wird, für jede einzelne Transaktion eine potenzielle Schadensschätzung abgegeben – unter der Annahme, dass es sich um einen Betrugsfall handelt. Dieses Regressionsmodell ergänzt die Fraud-Klassifikation um eine **quantitative** Risikoeinschätzung, indem es für jede Transaktion die potenziell finanzielle Auswirkung im Betrugsfall prognostiziert.

Durch die Kombination beider Modellstufen – Betrugswahrscheinlichkeit (Klassifikation) und Schadenshöhe (Regression) – kann die Entscheidungslogik gezielter gestaltet werden: So kann beispielsweise auch bei geringer Klassifikationssicherheit, aber hoher erwarteter Schadenshöhe, eine manuelle Kontrolle sinnvoll sein. Umgekehrt kann bei geringer Schadensprognose und nur schwacher Tendenz zum Betrugsverdacht bewusst auf eine Kontrolle verzichtet werden.

Das kombinierte Modell würde somit im Idealfall eine kostenbewusste und risikoorientierte Entscheidungsstrategie, bei der Aufwand und potenzieller Nutzen besser gegeneinander abgewogen werden können, bieten.

Auch im Regressionskontext hat sich gezeigt, dass Boosting-Algorithmen (z. B. LightGBM, XGBoost) für den vorliegenden Anwendungsfall die besten Ergebnisse liefern, weshalb im Folgenden nur ein XGBoost-Regressor verwendet wird.

Datengrundlage

Für die Schadensschätzung war zunächst nicht eindeutig, welcher Datensatz am besten zur Modellierung geeignet ist. Grundsätzlich standen drei Varianten zur Auswahl:

1. Der vollständige Datensatz – enthält sowohl betrügerische als auch reguläre Transaktionen, inkl. solcher mit Schaden = 0.
2. Ein ausgewogener Datensatz – mit einer gleichen Anzahl an betrügerischen (positiven) und unauffälligen (negativen) Fällen.
3. Ein eingeschränkter Datensatz – ausschließlich bestehend aus Fällen mit einem tatsächlichen Schaden (damage > 0).

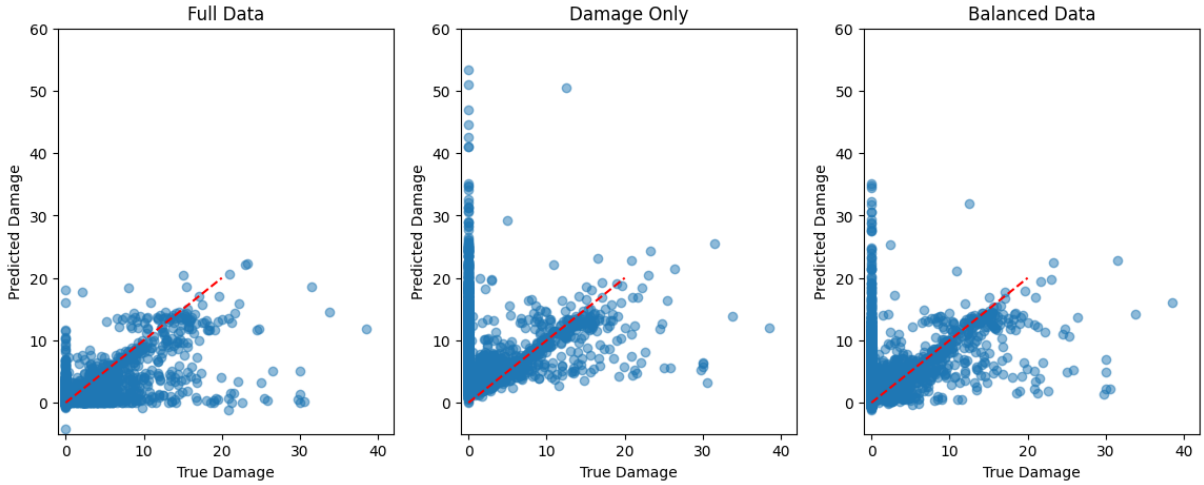
Diese Varianten unterscheiden sich hinsichtlich Zielsetzung und Relevanz für die spätere Modellanwendung:

- Variante 3 (nur Schadensfälle) erscheint zunächst sinnvoll, wenn der Fokus ausschließlich auf der genauen Schätzung der Schadenshöhe bei Betrugsfällen liegt. Allerdings kann ein Modell, das ausschließlich auf Schadensfälle trainiert wurde, später nicht sinnvoll mit regulären Transaktionen umgehen, da es diese nie gesehen hat.
- Variante 1 (kompletter Datensatz) erlaubt dem Modell, auch unauffällige Fälle korrekt als „kein Schaden“ zu erkennen, stellt es aber vor die Herausforderung, in einem stark unausgewogenen Zielverteilungsraum (viele Nullen) zu lernen.

- Variante 2 (ausgewogener Datensatz) bietet einen Kompromiss: Das Modell sieht sowohl Schadens- als auch Nicht-Schadensfälle in gleicher Anzahl, was zu einer besseren Sensitivität führen kann, allerdings ohne die reale Verteilung widerzuspiegeln.

Um diesen Zielkonflikt systematisch zu adressieren, wurden alle drei Varianten umgesetzt und miteinander verglichen, um die Auswirkungen auf die Modellgüte und Praxistauglichkeit zu bewerten.

Resultate



Tendenziell haben alle drei Varianten Schwierigkeiten, FRAUD-Fälle mit höheren Schadenssummen korrekt einzuschätzen. Im Bereich zwischen 0 und 10 zeigt sich eine hohe Streuung, wobei die Punkte nahezu quadratisch verteilt sind. Auffällig ist, dass die Variante, die ausschließlich auf FRAUD-Fällen trainiert wurde, normalen Transaktionen teilweise sehr hohe Schadenswerte zuweist. Im Bereich oberhalb von 10 gruppieren sich die Punkte bei dieser Variante jedoch am stärksten um die Ideallinie, was auf eine bessere Einschätzung bei hohen Schäden hindeutet.

Die traditionellen Metriken bieten hier keinen wirklichen Vergleich, seien der Vollständigkeit halber aber mit angegeben.

```
Metrics: Full data training set evaluation:
MSE                         1.06
RMSE                        1.03
MAE                         0.18
R2                          0.59
Metrics: Full data test set evaluation:
MSE                         1.12
RMSE                        1.06
MAE                         0.20
R2                          0.51

  Balanced data training set evaluation:
  MSE                         9.02
  RMSE                        3.00
  MAE                         1.55
  R2                          0.73
  Balanced data test set evaluation:
  MSE                         5.54
  RMSE                        2.35
  MAE                         1.22
  R2                         -1.41
```

```
Damage only training set evaluation:
MSE                     10.67
RMSE                     3.27
MAE                      1.95
R2                       0.74
Damage only test set evaluation:
MSE                     39.42
RMSE                     6.28
MAE                      5.73
R2                     -16.13
```

Die Testdaten waren für alle drei Modelle identisch (nur geringer Teil von FRAUD-Fällen), wodurch sich die negativen R2-Werte erklären lassen.

Vorhersagegüte der Schadenshöhe für erkannte Fraud-Fälle

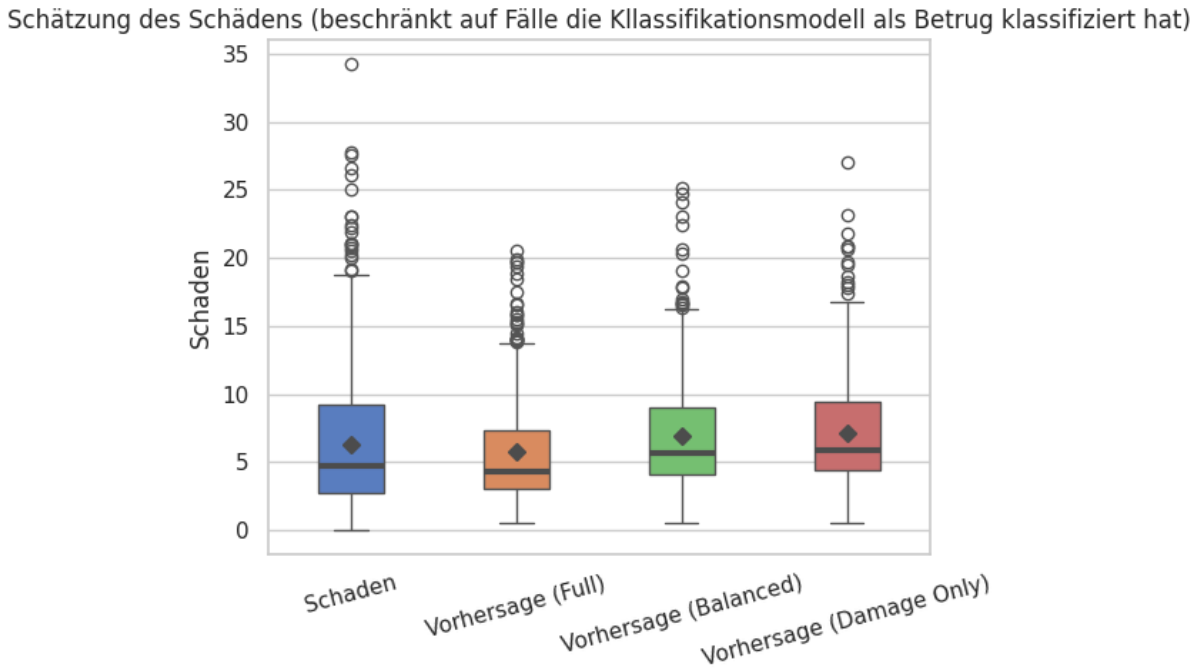
Neben ihrer Funktion als Entscheidungsgrundlage innerhalb des Modells soll die geschätzte Schadenshöhe auch extern über die Schnittstelle zur Verfügung gestellt werden. Ein naheliegender Anwendungsfall ist beispielsweise die Priorisierung erkannter Betrugsfälle nach wirtschaftlicher Relevanz.

Um die Qualität der Regressionsprognose in diesem konkreten Kontext zu evaluieren, wurde ein gezieltes Experiment durchgeführt: Es wurde untersucht, wie gut das Regressionsmodell den tatsächlichen Schaden für jene Transaktionen vorhersagt, die sowohl als tatsächliche Betrugsfälle vorliegen als auch korrekt vom Klassifikationsmodell als Fraud erkannt wurden.

Die Analyse beschränkt sich somit auf den relevanten Ausschnitt der Fälle, bei denen das Modell eine Kontrolle empfehlen würde und bei denen ein tatsächlicher Schaden entstanden ist. Dies liefert eine realitätsnahe Einschätzung der Prognosegüte in operativ relevanten Situationen.

Variante	R2	MAE	RMSE
full	0.466	2.679	4.185
balanced	0.411	2.759	4.389
damage_only	0.372	2.819	4.532

- Die Variante, die auf dem vollständigen Datensatz trainiert wurde, liefert mit einem R^2 von 0,466 die beste Vorhersagegüte. Das Modell erklärt somit etwa 47 % der Varianz der tatsächlichen Schadenshöhe in den relevanten Fällen.
- Gleichzeitig zeigt es auch die niedrigsten Fehlerwerte (MAE = 2,68 €, RMSE = 4,19 €) im Vergleich zu den anderen beiden Varianten.
- Das Modell, das ausschließlich auf Betrugsfällen mit Schaden trainiert wurde (“damage-only”), schneidet in allen Metriken am schlechtesten ab. Es scheint, dass diese eingeschränkte Perspektive die Generalisierungsfähigkeit des Modells auf reale Betrugsfälle reduziert.
- Die ausgewogene Variante (gleiche Anzahl positiver und negativer Fälle) liegt erwartungsgemäß dazwischen, bietet jedoch keinen Vorteil gegenüber dem vollständigen Trainingssatz.



Die Analyse des Boxplots zeigt deutliche Unterschiede zwischen den modellierten und den tatsächlichen Schadenshöhen. Die Variante, die auf dem vollständigen Datensatz trainiert wurde, neigt dazu, die Schadenshöhe systematisch zu unterschätzen. Im Gegensatz dazu überschätzen die beiden anderen Varianten („balanced“ und „damage-only“) den Schaden tendenziell. Ausserdem weisen alle drei eine signifikant geringere Varianz auf, was schon aus den R2-Werten deutlich wurde. Die tatsächliche Schadensverteilung zeigt eine deutlich höhere Streuung, insbesondere im oberen Bereich.

Vorhersage der Bewertungsfunktion

Ein möglicher Verwendungszweck des Regressionsmodells besteht darin, das Abschneiden des Klassifikationsmodells in Bezug auf die vorgegebene Bewertungsfunktion vorherzusagen. Dabei wird die potenzielle Schadenshöhe für jede Transaktion geschätzt – unabhängig davon, ob sie als Betrugsfall klassifiziert wurde oder nicht.

Anhand dieser Schätzung können zwei Szenarien verglichen werden:

1. Keine Kontrolle: Der geschätzte Schaden (falls es sich tatsächlich um Betrug handelt) bleibt bestehen.
2. Durchführung einer Kontrolle: prognostizierte Wahrscheinlichkeit * Bonus für entdeckten Fraud + (1 – Wahrscheinlichkeit) * Strafe für unnötige Kontrolle

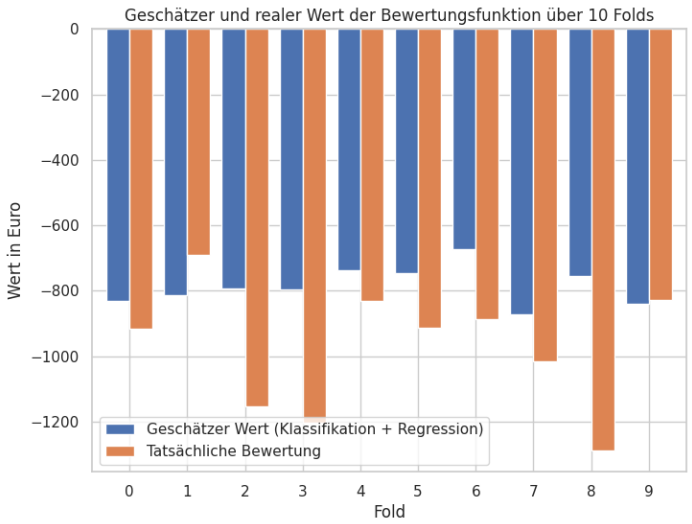
Durch den Vergleich beider Szenarien lässt sich für jede Transaktion entscheiden, ob sich eine Kontrolle wirtschaftlich lohnt. Diese Entscheidung wird für alle Transaktionen im Trainingsdatensatz simuliert, um das Zusammenspiel von Klassifikator und Regressor im Hinblick auf die Bewertungsfunktion ganzheitlich zu bewerten.

Als Baseline wurde das Klassifikationsmodell an sich mit einem konstanten Schaden in Höhe des Mittelwerts über alle Transaktionen ausgewertet. Das Modell kann die Bewertungsfunktion nur sehr schlecht vorhersagen und die Werte liegen um ein Vielfaches unter dem tatsächlichen Wert.

Geschätzer Wert (nur Klassifikationsmodell)	-4325.20
Geschätzer Wert (Klassifikation + Regression)	-831.48
Tatsächliche Bewertung	-917.32
enthaltene Schadenssumme	5973.03

In Kombination mit dem Regressionsmodell gelingt es, die tatsächliche Ausprägung der Bewertungsfunktion auf dem Testdatensatz insgesamt überraschend gut zu approximieren. Hierfür wurde die Variante des Regressionsmodells, das auf allen Trainingsdaten gelernt hat verwendet. Es hat sich zeigt, dass nur eine der getesteten Trainingsvarianten in der Lage ist, die Bewertungsfunktion realistisch abzubilden. Die beiden alternativen Ansätze – das Training ausschließlich auf Schadensfällen bzw. auf dem ausgewogenen Datensatz – erwiesen sich für diesen Anwendungszweck als gänzlich ungeeignet. Die vorhergesagten Werte der Bewertungsfunktion liegen bei diesen Modellen um mehrere Größenordnungen tiefer als die tatsächlichen Werten. Dies erklärt sich durch den signifikant höheren Mittelwert (5.1 bzw. 4.7 für die Varianten nur FRAUD-Fälle / ausbalanciert gegenüber 0.21 bei der Variante, die auf den vollständigen Daten trainiert wurde).

Das verwende Modell neigt jedoch dazu, sein erwartetes Abschneiden tendenziell zu optimistisch einzuschätzen.



Abschließend lässt sich festhalten, dass das Regressionsmodell, das auf dem vollständigen Datensatz trainiert wurde, in beiden Evaluationsdimensionen – sowohl bei der allgemeinen Vorhersagegüte als auch bei der spezifischen Schätzung der Schadenshöhe in korrekt erkannten Betrugsfällen – konsistent bessere Ergebnisse liefert als die Varianten mit balancierten oder ausschließlich auf Schadensfällen basierenden Trainingsdaten. Trotz einer leichten systematischen Unterschätzung hoher Schadensbeträge bietet diese Variante die robusteste und stabilste Leistung über verschiedene Testszenarien hinweg und eignet sich daher am besten für die praktische Anwendung im Gesamtsystem.

Stufe 4: Bewertung der Handlungsalternativen

In dieser Stufe soll auf Basis der beiden Modellprognosen – der Betrugswahrscheinlichkeit und der geschätzten Schadenshöhe – eine wirtschaftlich begründete Entscheidung getroffen werden: Soll eine Transaktion kontrolliert werden oder nicht? Die Entscheidung basiert auf der folgenden, aus der Bewertungsfunktion abgeleiteten Entscheidungslogik. gain_tp steht dabei für den wirtschaftlichen Gewinn bei einem entdeckten Betrugsfall (hier: 5 € laut Definition der Wertkauf GmbH), cost_fp hingegen für die Kosten, die durch eine unnötige Kontrolle entstehen (hier: 10 €).

Alternative 1: Kontrolle wird durchgeführt:

Erwarteter Gewinn bei Kontrolle: *P(FRAUD)* * gain_tp – *P(NORMAL)* * cost_fp*

Alternative 2: Kontrolle wird **nicht** durchgeführt:

Erwarteter Verlust bei Nicht-Kontrolle: *P(FRAUD) * Erwarteter Schaden*

Entscheidungsregel: Kontrollieren, wenn erwarteter Gewinn > erwarteter Verlust

Da nicht von vornherein klar ist, welche Variante des Regressionsmodel am besten für diese Aufgabe geeignet ist, wurden in gewohnter Manier Tests mit 5-facher Kreuzvalidierung bei 5 Wiederholungen mit allen drei Varianten ausgeführt. Es wurden dabei folgenden Varianten und Kombinationen untersucht.

- Die Entscheidungsregel kommt nur zum Tragen, wenn die vom Klassifikationsmodell errechnete Wahrscheinlichkeit über dem Schwellwert von 0.5 liegt.
- Es wird in jedem Fall nach der Entscheidungsregel entschieden.

Die Variante, die Entscheidungsregel nur bei als FRAUD klassifizierten Fällen einzusetzen, liefert für alle 3 Varianten des Regressionsmodels schlechtere Ergebnisse. Die Tabelle zeigt die Differenz zum Baseline Modell (reine Klassifikation):

	Mittlere Differenz der Bewertung (mit Threshold)
Full	-15.535
Balanced	-11.178
Damage Only	-3.536

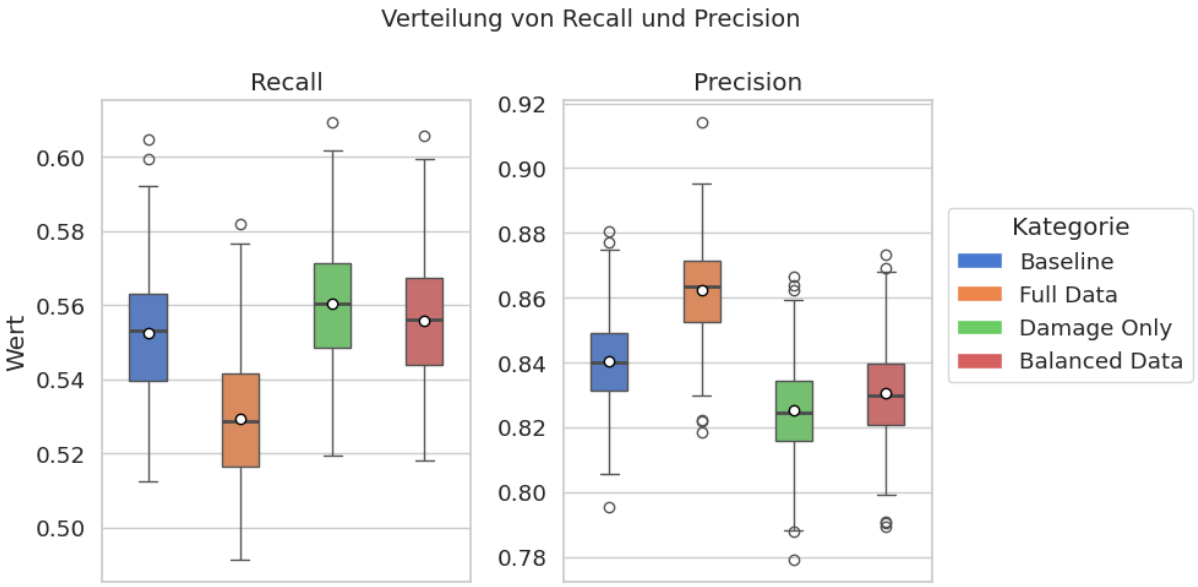
Auch die zweite Variante schneidet in der Bewertung nicht wirklich besser ab, als das reine Klassifikationsmodell.

	Mittlere Differenz der Bewertung
Full	-4.92
Balanced	7.71
Damage Only	4.02

Die auf dem vollständigen Datensatz trainierte Regressionsvariante ("Full") zeigt im Vergleich zu den Varianten, die entweder ausschließlich auf Schadensfällen oder auf einem ausbalancierten Datensatz basieren, ein im vorliegenden Anwendungsfall vorteilhafteres Verhältnis von Precision zu Recall.

Während die alternativen Varianten eine minimal bessere Gesamtbewertung erzielen, geschieht dies primär durch einen höheren Recall. Das heißt, es werden mehr tatsächliche Fraud-Fälle erkannt. Allerdings geht dieser Zugewinn an Sensitivität mit einer geringeren Precision einher, was in der Praxis zu mehr unnötigen Kontrollen führt.

Die „Full“-Variante erzielt eine vergleichbare Gesamtbewertung, jedoch mit einer höheren Precision und damit einer geringeren Anzahl an falsch-positiven Fällen. In der praktischen Umsetzung bedeutet dies, dass weniger unberechtigte Verdachtsfälle zu einer Kontrolle führen. Da solche Fehlalarme potenziell zu einer negativen Kundenerfahrung führen und auch zusätzlichen personellen Aufwand verursachen, ist dieser Aspekt insbesondere unter wirtschaftlichen und serviceorientierten Gesichtspunkten als klarer Vorteil zu bewerten.



Mehrwert des Modells

Die folgende Auswertung basiert auf einer umfangreichen Evaluation des Modells über 200 unabhängige Testläufe (5-fache Kreuzvalidierung mit 40 Wiederholungen). Durch diese robuste Evaluationsstrategie wird sichergestellt, dass der beobachtete durchschnittliche wirtschaftliche Mehrwert nicht auf Zufall oder spezifische Datensplits zurückzuführen ist, sondern als belastbare Schätzung der zu erwartenden Modellleistung gilt.

Die Anwendung des kombinierten Modells aus Klassifikation, Regressionsschätzung und Entscheidungslogik führt zu einem durchschnittlichen wirtschaftlichen Mehrwert von 0,22 € pro Transaktion. Dieser Wert berücksichtigt die drei Faktoren, die in die Bewertungsfunktion einfließen und stellt somit den tatsächlichen ökonomischen Mehrwert gemäß Bewertungsfunktion dar. Die folgenden Angaben beziehen sich hierbei auf den Testdatensatz (entsprechend 20 Prozent aller zur Verfügung gestellten Datensätze).

	Schaden	Bewertung	Mehrwert	Anzahl Transaktionen
Model	-6015.16	-1005.27	5009.90	29604.8
Statische Regeln	-1057.82	393.00	1450.82	78.6
Gesamt	-7072.98	-612.27	6460.71	29683.4

Pro Transaktion entspricht das einem Mehrwert von 0.22 Euro.

Betrachten wir nur den Schaden, den das Modell verhindert, sieht die Situation wie folgt aus:

	Schaden verhindert	Anzahl Transaktionen
Model	3468.27	29604.8
Statische Regeln	1057.82	78.6
Gesamt	4526.09	29683.4

Betrachtet man nur den verhinderten Schaden, also ohne Abzug der Kontrollkosten und ohne Bonus für entdeckte FRAUD-Fälle, ergibt sich ein mittlerer Wert von 0,15 € pro Transaktion.

Betrachtung: Schaden durch Rabatt

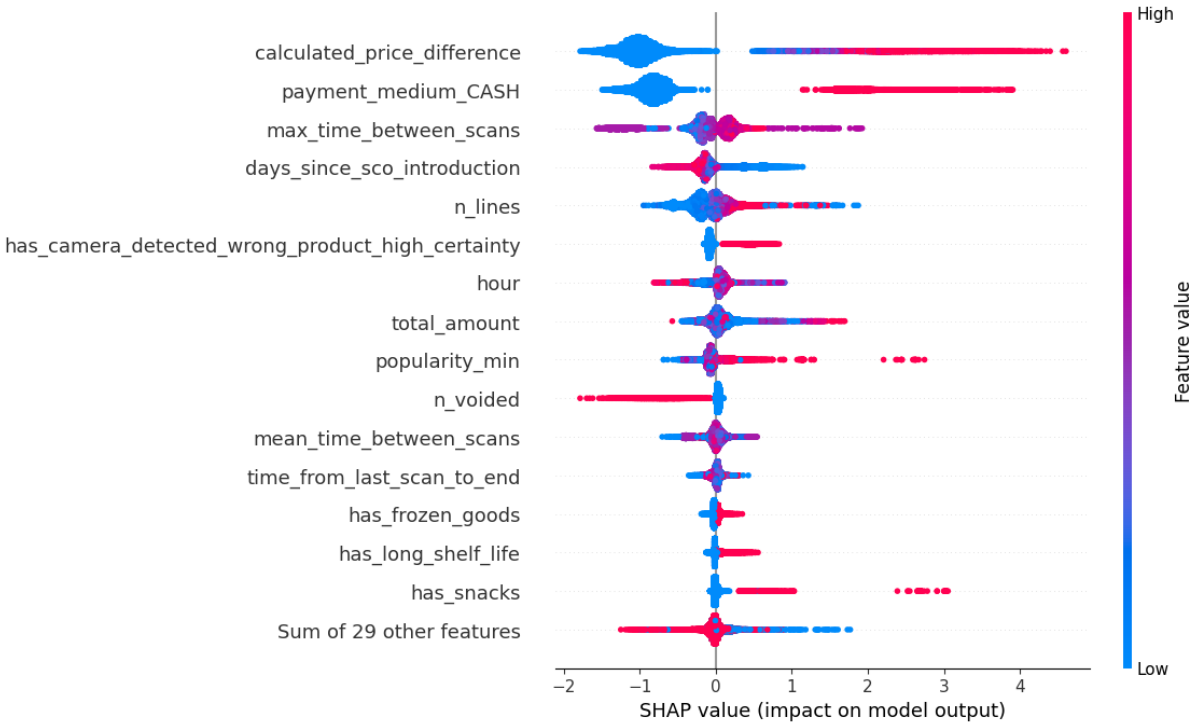
Eine Analyse mit den gewohnten Methoden (Kreuzvalidierung und Wiederholung) ergab, dass das Modell im Schnitt 62% der unberechtigten Rabattfälle erkannt hat und somit einen Anteil von 63% des entstandenen Schadens verhindern würde.

```
Anzahl Fälle mit Rabatt und Fraud      580.80
Anzahl korrekt vorhergesagter Fälle    359.70
Anteil erkannt                           0.62

Schaden der Fälle mit Rabatt und Fraud      3938.94
Schaden der korrekt vorhergesagten Fälle    2462.49
Anteil Schaden verhindert                      0.63
```

Sensitivitätsanalyse

Die Sensitivitätsanalyse des Klassifikationsmodells zur Betrugserkennung an Selbstbedienungskassen zeigt deutlich, welche Merkmale das Modell maßgeblich zur Entscheidung heranzieht. Einige Zusammenhänge wurden bereits in Meilenstein 2 beleuchtet und konnten im Rahmen von Meilenstein 3 nochmals, auch bei den komplexeren Modellen, bestätigt werden. Dazu zählt beispielsweise die besondere Rolle des Zahlungsmittels: Die Verwendung von Bargeld ist im Modell mit einem deutlich erhöhten Betrugsrisiko assoziiert und stellt in vielen Fällen den stärksten Einzelindikator dar.

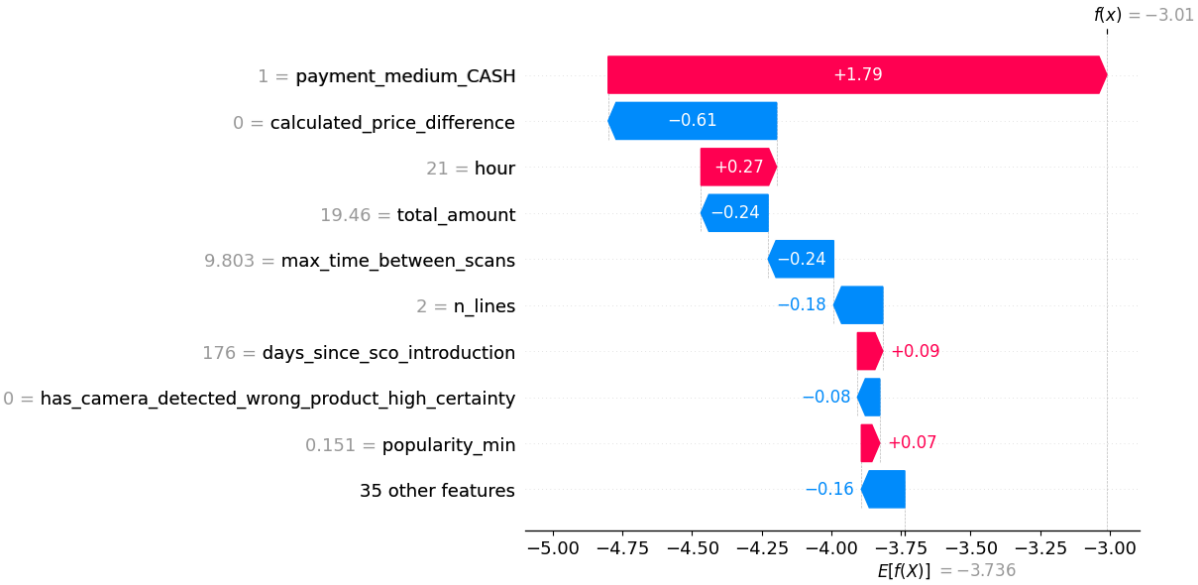


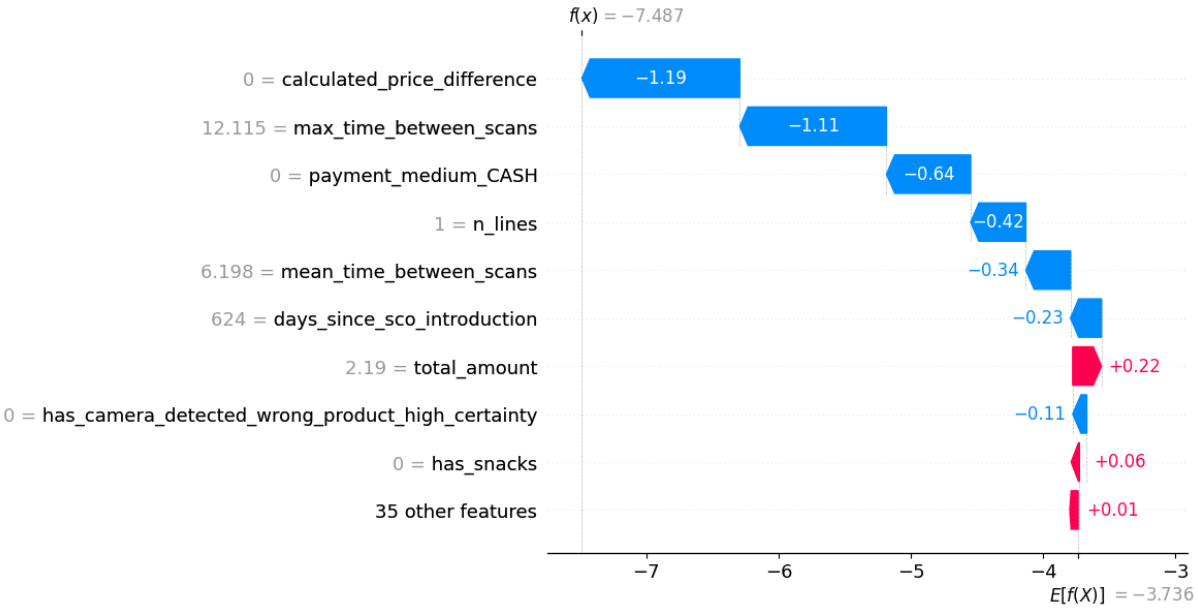
Auch technisch messbare Unregelmäßigkeiten wie ein abweichender Preis (calculated_price_difference) oder ein durch Kameras detektierter falscher Produktscan mit hoher Sicherheit wirken stark risikoerhöhend. Diese Merkmale werden vom Modell konsistent als typische Muster betrügerischen Verhaltens bzw. von Fehlern bei der Transaktion erkannt.

Zeitbasierte Merkmale wie die maximale Dauer zwischen zwei Scans oder die Tageszeit des Einkaufs tragen ebenfalls zur Bewertung bei, jedoch mit geringerem Einfluss auf die Klassifikation. So führt ein zögerliches Scannen tendenziell zu einer Erhöhung der Fraud-Wahrscheinlichkeit, allerdings in Kombination mit anderen Faktoren. Wie bereits in der Datenanalyse im letzten Meilenstein diskutiert, zeigt die Analyse, dass auch die Dauer des im Einsatz befindlichen Kamerasystems (days_since_sco_introduction) eine Rolle spielt – neu eingeführte Kamerasysteme sind weniger aussagekräftig als länger im Einsatz befindliche.

Exemplarisch können einzelne Merkmalswerte und deren Kombination aufzeigen, wie weit der Algorithmus bei sonst gleichen übrigen Merkmalsausprägungen seine Klassifikation in Richtung NORMAL bzw. FRAUD verändern würde, das Modell seine Risikoeinschätzung verschiebt. Das

Modell liefert eine inhaltlich nachvollziehbare Entscheidungssystematik, die über eine reine Blackbox hinausgeht. Das durchschnittliche FRAUD-Risiko (bezogen auf das gesamte Modell) beträgt -3.736 und wird im ersten Transaktions-Beispiel insbesondere durch die Barzahlung auf – 3.01 erhöht, während sich dieses Risiko im zweiten Beispiel durch die aufgeführten Einflussgrößen auf -7.487 verringert.





Handlungsempfehlungen

Das erhaltene Gesamtmodell aus statischen Vorabfiltern sowie die anschließende Klassifikation und Regression zeigen insgesamt eine sehr überzeugende Güte in der Entdeckung von Betrugsfällen und sollte die aktuellen diskretionären Kontrollen ablösen. So können nachhaltig Schäden verhindert (durch das Aufspüren von Betrugsfällen) und Kosten minimiert (durch effizienten Einsatz von Personal) werden. In den vorangegangenen Kapiteln wurden die Einzelheiten des Modells und Details zum verhinderten Schaden ausführlich dargestellt.

Bei sich verändernden Bedingungen (z.B. eine neue Filiale oder Einbindung von Filialen mit gänzlich anderem Warenangebot bzw. einer anderen Kundschaft als in den Trainingsdaten) sollte der Algorithmus auf diese neuen Daten entsprechend durch Rekalibrierung vorbereitet werden. Ein Algorithmus lernt aus historischen Daten und nur bei ausreichender Ähnlichkeit von historischen Trainingsdaten im Vergleich zu Evaluationsdaten kann eine im Training erreichte Güte später ebenfalls im Livebetrieb erreicht werden. Ebenso kann zukünftig die Güte des Modells weiter verbessert werden, wenn unterstützende Systeme (wie z.B. das Kamerasystem) zusätzliche Daten liefern oder durch die zukünftig effizienter durchgeführten Kontrollen neue Trainingsdaten verfügbar sind. Das hier vorgestellte Basismodell kann auf diese Weise stetig verbessert werden.

Wie bereits erläutert, stellt die manuelle Eingabe von Rabatten durch Kundinnen und Kunden ein besonders auffälliges Risiko dar. Wie in der Analyse gezeigt, geht das Betätigen der Rabattfunktion in bestimmten Fällen mit einer deutlich erhöhten Betrugswahrscheinlichkeit einher.

Aus diesem Grund empfiehlt es sich, bereits auf technischer Ebene präventive Maßnahmen zu prüfen. Dazu zählen etwa:

- das gezielte **Deaktivieren oder Ausblenden der Rabattfunktion** bei Produkten oder Produktkategorien, für die typischerweise keine Rabattgewährung vorgesehen ist (z. B. Haushaltswaren), oder
- die Verwendung **vordefinierter Rabatt-Barcodes** (z. B. auf Stickern), die den Preisnachlass automatisch und kontrolliert auslösen.

Ergänzend dazu bietet es sich an, im Rahmen der Modellierung **statische Regeln für risikobehaftete Rabattkonstellationen** zu definieren, um derartige Fälle frühzeitig zu erkennen und gezielt zu kontrollieren.

Ausblick

Technische Implementierung und REST-Schnittstelle

Das Modell wird der Wertkauf GmbH über eine REST-Schnittstelle zur Verfügung gestellt und kann in das Kassensystem der Wertkauf GmbH integriert werden. Diese erlaubt:

- Entgegennahme von Transaktionsdaten im JSON-Format
- Entscheidung „Kontrolle: Ja/Nein“ in Echtzeit mit entsprechender Begründung und einer Schätzung des verhinderten Schadens durch diese Entscheidung.

Der vollständige Programmcode befindet sich im GitHub Repository, sodass hier auch die Möglichkeit besteht, das Modell zukünftig bei veränderten Daten ganz oder teilweise neu zu trainieren.

Evaluierung auf Basis der Kundendaten

Es ist vorgesehen, das Modell und die getroffenen Maßnahmen anhand von Testdaten der Wertkauf GmbH zu evaluieren und ggf. weitere Optimierungen der Kontrollstrategie durchzuführen.

Langfristig ist die Einbindung weiterer Datenquellen (z. B. Kundenhistorie, Treuekarten, Warenkorbdaten) denkbar.