Abgabedatum: 22.06.2025

Projektgruppe: Raphael Schaffarczik, David Zurschmitten, Matthias Bald

Auftraggeber: Wertkauf GmbH

ANALYSE

**Verlustprävention an Selbstbedienungskassen im Einzelhandel**

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 Meilensteines ist darüber hinaus, dem Kunden Handlungsempfehlungen aufzuzeigen, den Werteverlust an seinen Selbstbedienungskassen zu reduzieren und gleichzeitig nicht notwendige Kontrollen zu vermeiden.

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 konkreten Projekt werden folgende Ziele verfolgt:

* Ableitung statischer Regeln zur sofortigen Klassifikation offensichtlicher Betrugsfälle
* Entwicklung eines Klassifikationsmodells zur Vorhersage der Betrugswahrscheinlichkeit
* Entwicklung eines Regressionsmodells zur Schätzung der potenziellen Schadenshöhe bei Betrug
* Integration beider Modelle in eine ökonomisch fundierte Entscheidungsfunktion
* Durchführung einer Schwellenwert- und Sensitivitätsanalyse zur Optimierung der Kontrollstrategie
* 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, liegen valide und repräsentative Merkmalsräume für die Modellbildung vor.

Die vorliegenden Daten basieren auf rund 150.000 Transaktionen aus einem Echtbetrieb von Selbstbedienungskassen im Lebensmitteleinzelhandel. Jeder Datensatz beschreibt einen abgeschlossenen Kaufvorgang und enthält Informationen zu Bezahlvorgang, Artikeln, Uhrzeit, Kameradaten, sowie manuelle oder automatische Rückmeldungen von Kunden und Systemen. 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 R² 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 Merkmalskombinationen 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 Aussortieren von Merkmalen wäre daher kontraproduktiv gewesen und hätte potenziell relevante Informationen ausgeschlossen.

|  |  |
| --- | --- |
| **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 |
| 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 |
| Produktspezifische Merkmale |  |
| 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 |
| 9. Standortmerkmale / Filialdaten | store\_id, location, urbanization |

Anforderungen an Analyseverfahren

Bei der Umsetzung des dritten Meilensteins „Analyse“ nach DASC PM ergeben sich bestimmte Anforderungen, die einen Einfluss auf die Auswahl eines geeigneten Modells bzw. Analyseverfahren haben.

Im Zusammenhang mit dem konkreten Projekt der Verlustprävention an Selbstbedienungskassen ergeben sich die folgenden wesentlichen Merkmale bzw. Kriterien:

|  |  |
| --- | --- |
| 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** | **% FRAUD** | **Schaden gesamt** |
| --- | --- | --- | --- | --- | --- |
| 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 Mindesthalbarkeitsdatum). 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.

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

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.

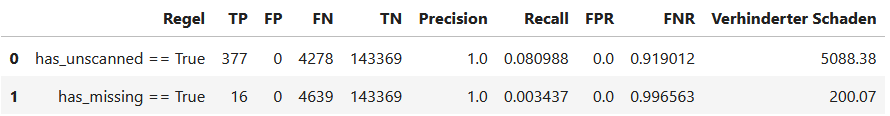
Zentrale Attribute bei der Regelbildung waren has\_unscanned und has\_missing. Es handelt sich um Transaktionen, bei denen von der Kamera fehlende Positionen erkannt wurden und automatisch in die Transaktion als FRAUD aufgenommen wurden (has\_unscanned) bzw. Transaktionen mit einer fehlenden Produktidentifikation.

**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.



**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]](#footnote-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.

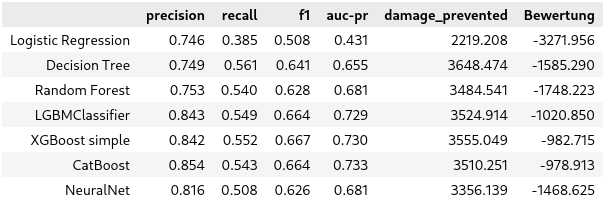
Trotz wiederholter Versuche und verschiedener Architekturvarianten war die Konsistenz der Trainingsverläufe nicht ausreichend gewährleistet.

**Modellvergleich:**

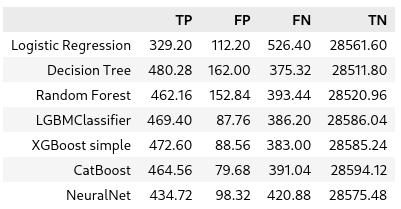
* Datengrundlage: für das Training und Evaluation wurden die gelabelten Daten, ohne die durch die statischen Regeln (has\_unscaned, has\_missing) klassifizierbaren Transaktionen gewählt
* Für das Lineare Modell wurden nur die 3 in Phase 2 als relevant bestimmten Merkmale verwendet
* Die übrigen Modelle 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:**

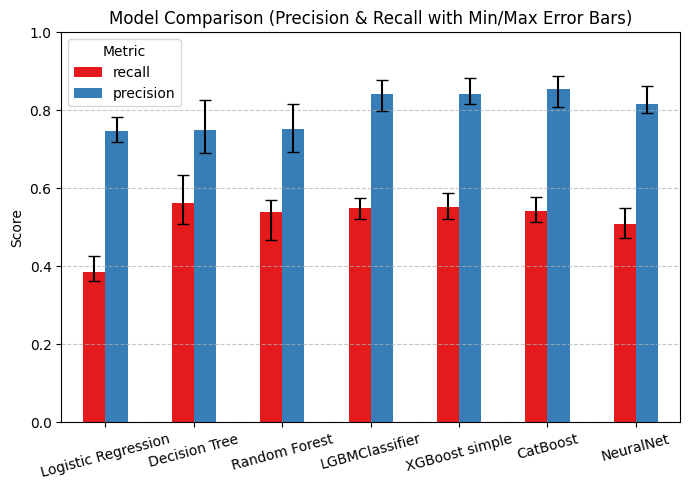
Arithmetische Mittelwerte ausgewählter Metriken über alle Testruns:



Arithmetische Mittelwerte der Konfusionsmatrizen über alle Testruns:

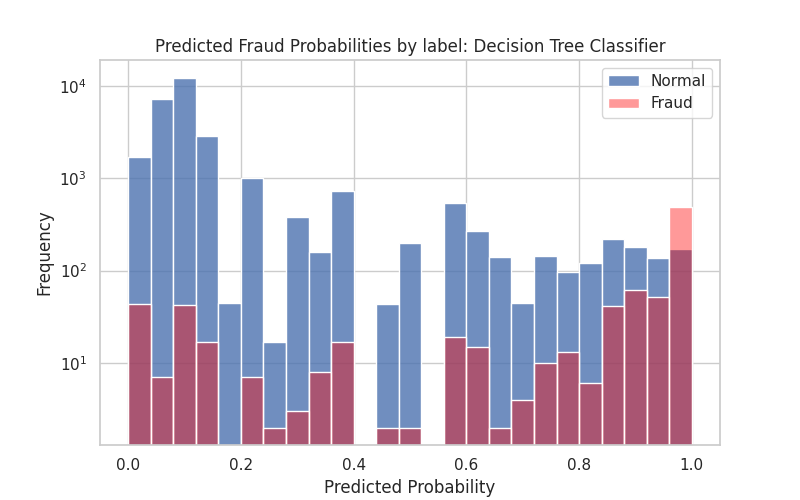
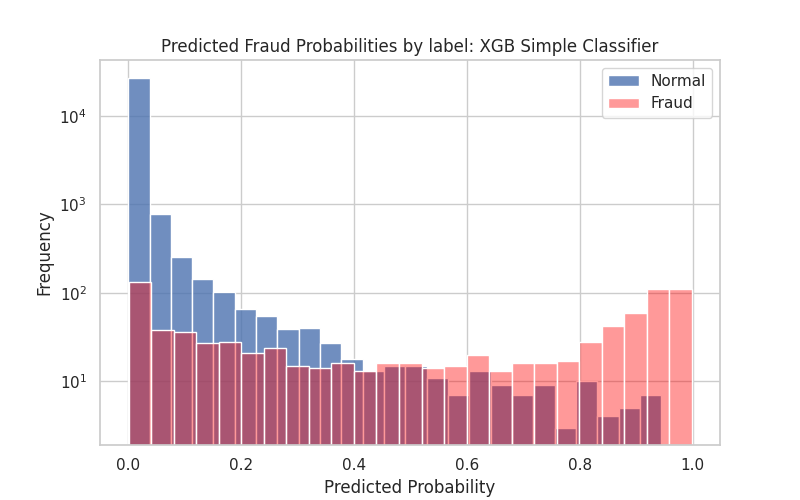
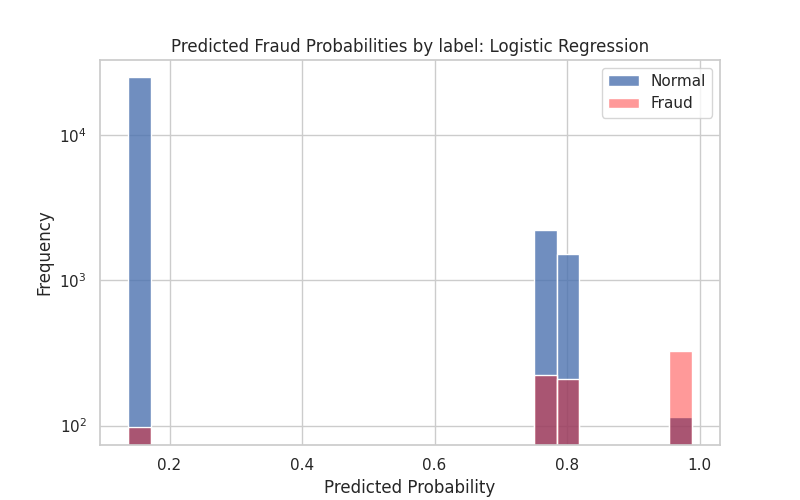


Die Boosting-Modelle (LightGBM, XGBoost, CatBoost) liefern die insgesamt besten Ergebnisse sowohl in der Klassifikationsgüte als auch in der wirtschaftlichen Relevanz. Sie überzeigen mit einer hohen Präzision (wenig unnötige Kontrollen) und erkennen ca. 55% der Fraud-Fälle. Das sind 5% mehr als das neuronale Netz, das ebenfalls ein 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 (insbesonder 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.



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.

**Schwellwertoptimierung:**

Wie bereits angerissen, können die Score Werte des Klassifikationsmodells nicht direkt als Wahrscheinlichkeiten interpretiert werden, sofern sie nicht ordnungsgemäß kalibriert wurden. Manche Modelle kalibrieren den Output von vornherein gut (z.B. klassisiche 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 Outputwerte angepasst werden.

Zusätzlich ergibt sich noch ein weiterer Punkt: Die Modelle wurden auf anderen Kostenfunktionen traininert als die später zu evaluierende Bewertungsfunktion des Kunden. Hier ergibt sich erneut eine Schwellwertoptimierung anhand der vom Kunden vorgegebenen Strafen bzw. Belohnungen für richtige bzw. falsche Klassifikationen.

Beide Punkte werden daher zusammengefasst im späteren Kapitel als gemeinsame Schwellwertoptimierung behandelt.

Stufe 3: Regressionsmodell zur Schadensschätzung

Ziel dieser Modellkomponente ist es, die finanzielle Schadenshöhe zu prognostizieren, die im Falle eines nicht erkannten Betrugsfalls 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 potenzielle 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).

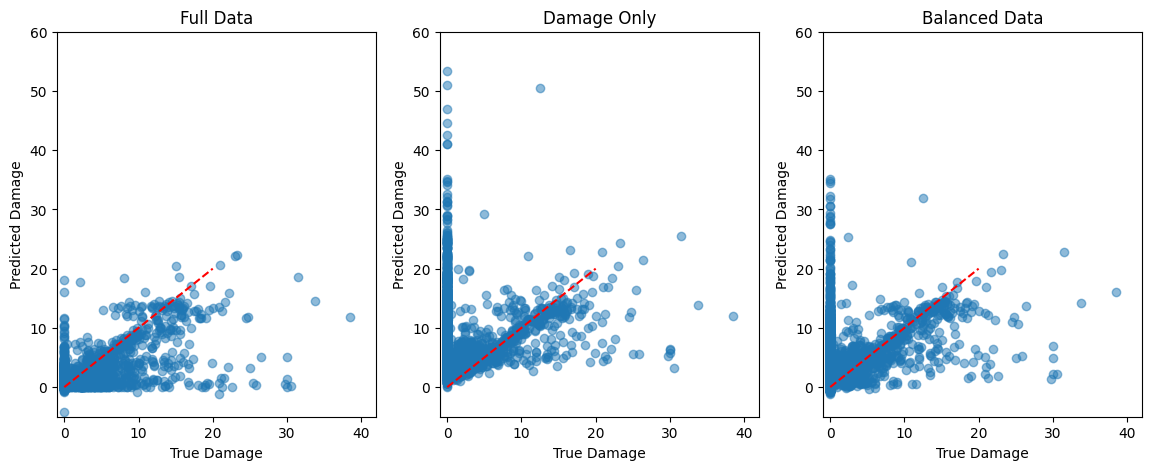
Diese Varianten unterscheiden sich hinsichtlich Zielsetzung und Relevanz für die spätere Modellanwendung:

* 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.

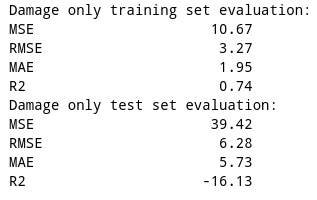
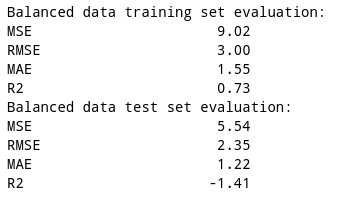
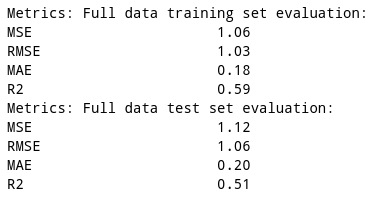
**Resultate**

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



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 natürlich keinen wirklichen Vergleich, seien der Vollständigkeit halber aber mit angegeben.



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

**Vorhersage der Bewertungsfunktion**

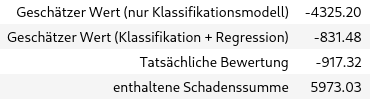
Ein möglicher Verwendungszweck des Regressionsmodells besteht darin, das Abschneiden des Klassifikationsmodells in Bezug auf die vorgegebene Bewertungsfunktion vorherzusagen. Dabei wird die potenzielle Schadenshöhe für jede Transaktion geschätzt – unabhängig davon, ob sie als Betrugsfall klassifiziert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

Anhand dieser Schätzung können zwei Szenarien verglichen werden:

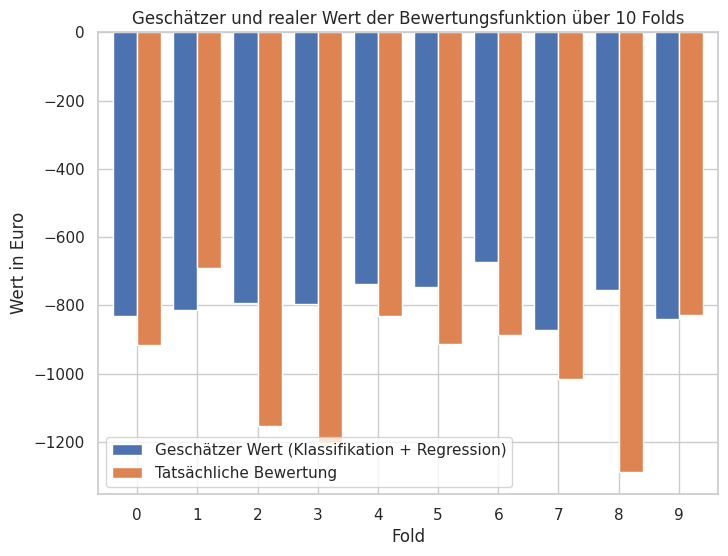
1. Keine Kontrolle: Der geschätzte Schaden (falls es sich tatsächlich um Betrug handelt) bleibt bestehen.
2. Durchführung einer Kontrolle: prognostizierte Wahrscheinlichkeit \* Bonus für entdeckten Fraud + (1 – Wahrscheinlichkeit) \* Strafe für unnötige Kontrolle

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 tatsälichen Wert.



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 – erwies sich für diesen Anwendungszweck als ungeeignet. Die vorhergesagten Werte der Bewertungsfunktion liegen bei diesen Modellen um mehrere Größenordnungen neben den tatsächlichen Werten Das Modell neigt jedoch dazu, sein erwartetes Abschneiden tendenziell zu optimistisch einzuschätzen.



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? Als Grundlage für die Entscheidung dient die folgende auf der Bewertungsfunktion basierende Entscheidungsfunktion.

**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 einem Schwellwert liegt (unterschiedliche Schwellwerte: 0.5, 0.4, 0.3). **ACHTUNG: THEMA SCHWELLWERTOPTIMIERUNG FEHLT: SIEHE ABSCHNITT UNTEN (AUCH IN ROT)**
* Es wird in jedem Fall nach der Entscheidungsregel entschieden.
* Der Wert von *gain\_tp* in der Entscheidungsfunktion wird konsant als 5 angenommen (identisch zur Bewertungsfunktion).
* Im Fall von *cost\_fp* wurden zwei Varianten untersucht: erstens der fixe Wert 10 aus der Bewertungsfunktion, zweitens wurde auf den Trainingsdaten derjenige Wert für *cost\_fp* bestimmt, mit dem ein optimales Ergebnis erzielt wird. (Wichtig zu betonen: Die Bewertungsfunktion an sich wurde natürlich nicht angepasst, nur der Parameter in der Entscheidungsfunktion).

Wie im vorhergehenden Abschnitt gezeigt, ist das Modell bereits gut in der Lage, den zu erwartenden Wert der Bewertungsfunktion vorherzusagen. Die beobachtete Abweichung zwischen geschätztem und tatsächlichem Ergebnis lässt sich möglicherweise weiter reduzieren:

Eine gezielte Anpassung des Parameters cost\_fp, der die angenommenen Kosten für eine unnötige Kontrolle (False Positive) steuert, könnte dabei helfen, die Entscheidungsschwelle besser auszubalancieren und die verbleibende Überschätzung zu verringern. Dies würde die Prognosegenauigkeit des kombinierten Modells weiter verbessern und zu stabileren Entscheidungen im operativen Einsatz führen..

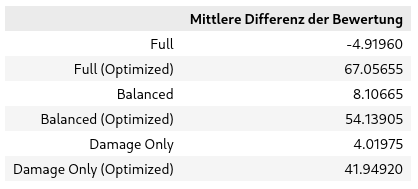
Im Ergebnis hat sich gezeigt, dass die Verwendung des fixen Wertes 10 in der Entscheidungsfunktion bei allen Schwellwerten bzw. auch ohne Schwellwerten nur zu geringfügig besseren Ergebnissen als die Baseline führen (nur Klassifikationsmodell mit Schwellwert 0.5).

Wird *cost\_fp* als Parameter in der Entscheidungsfunktion definiert und optimiert, lassen sich bessere Ergebnisse erzielen. Grundsätzlich hat sich gezeigt, je geringer der Schwellwert, desto besser. Setzt man den Schwellwert auf 0 führen allen drei Varianten des Regressionsmodells zu besseren Ergebnissen. Am besten schneidet jedoch das Regressionsmodell ab, das auf dem vollen Datensatz trainiert wurde.

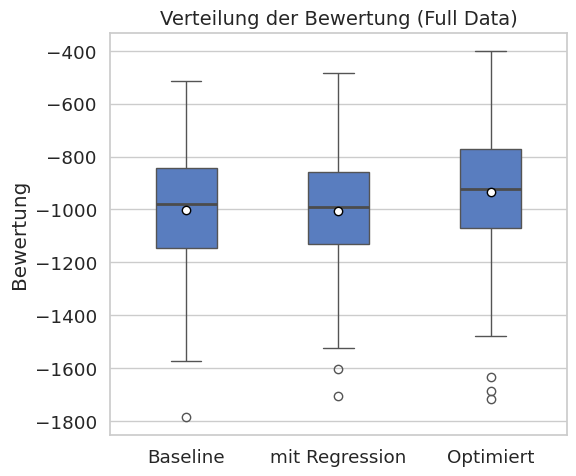
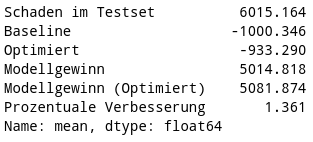
**Auswertung der Ergebnisse der besten Modellvariante**

Die Variante mit den besten Ergebnissen (Schwellwert 0 und Optimierung von *cost\_fp)* wurde nochmal in 200 Testläufen getestet (5-facher Kreuzvalidierung bei 40 Wiederholungen) um die Schätzung der Ergebnisse zu verbessern.

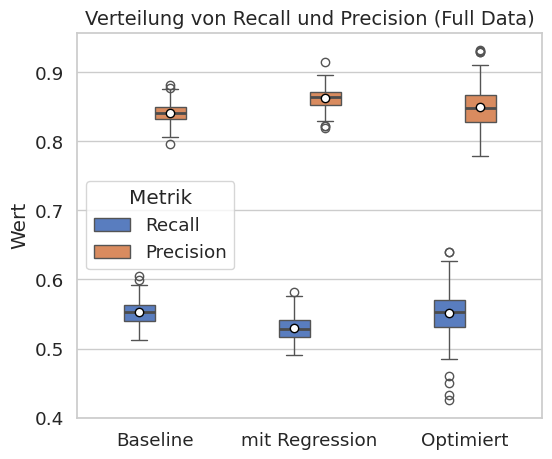
Die Tabelle zeigt die mittlere Differenz zwischen der erreichten Bewertung des kombinierten Modells (Klassifikation + Regression + Entscheidungsfunktion mit und ohne Optimierung des Parameters *cost\_fp* ) und der Baseline.



Das kombinierte Modell mit Optimierung erzielte im Mittelwert einen um 1.36 Prozent höheren Gewinn (Schaden in der Testmenge – Bewertung des Modells) als das Baseline-Modell.



Die Abweichungen in den Bewertungen über die 200 Testläufe sind bei allen drei Varianten ähnlich gelagert.



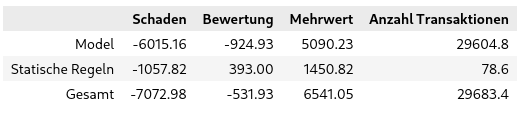
Die Baseline und die Variante mit Regression erzielen in etwa gleiche Bewertungen, aber auf unterschiedlichen Wegen. Das Modell mit Regression hat einen tieferen Recall, erkennt also weniger FRAUD-Fälle, macht das aber durch eine höhere Präzision gut, indem es weniger unnötige Kontrollen verursacht. Das optimierte Modell hat ähnliche Werte wie das Baseline-Modell. Die grössere Bandbreite erklärt sich wahrscheinlich dadurch, dass das Modell je nach Testet die besseren Bewertungen entweder durch einen höheren Recall oder eine höhere Präzision erreicht, je nachdem was in dem konkreten Testset besser funktioniert.

Zusammenfassend lässt sich sagen, dass das kombinierte Modell - insbesondere in der Variante mit Schwellenwert 0 und optimiertem *cost\_fp*-Parameter - im Vergleich zur Baseline eine leicht, aber konsistent bessere Performance erzielt **Anmerkung Raphael: Der gesamte gelb markierte Abschnitt sollte meiner Meinung nach gelöscht werden, da eine Optimierung der fp-Strafterms auf unser kalibriertes Modell inhaltlich keinen Sinn macht. Das müsste erst einmal ökonomisch begründet werden.**

**Anmerkung von Raphael: Hier fehlt meiner Meinung noch ein klares Statement zum Schwellwert des Klassifikatores (des „Score“ Wertes). Warum werden die in dem früheren Kapitel als Wahrscheinlichkeiten nicht verwertbaren Scores ohne Rekalibrierung / Schwellwertoptimierung 1:1 in unseren Formeln als Wahrscheinlichkeiten verwendet (da, wo es darum geht, die Handlungsalternativen zu bewerten)? Dieser Übergang – ein verbesserter Schwellwert ist nicht erreichbar, deshalb verwenden wir den Output des Klassifikators direkt – ist nicht klar dargestellt.**

Nutzen des Modells

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 gemässe Bewertungsfunktion dar.



Pro Transaktion entspricht das einem einem Nutzen von 0.22 Euro.

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



Betrachtet man nur den verhinderten Schaden, also ohne Abzug der Kontrollkosten und ohne Bonus für entdeckte FRAUD-Fälle, ergibt sich ein mittlerer Wert von 0,18 € pro Transaktion.

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.

Ein Bild, das Text, Screenshot, Reihe, Diagramm enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

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 wurde, das Modell seine Risikoeinschätzung verschiebt. Das Modell liefert eine inhaltlich nachvollziehbare Entscheidungssystematik, die über eine reine Blackbox hinausgeht.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Handlungsempfehlungen

Das erhaltene Gesamtmodell aus statischen Vorabfiltern sowie die anschließende Klassifikation und Regression zeigen insgesamt eine sehr überzeugende Güte in der Detektierung 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. Schrittweise kann sodass hier vorgestellte Basismodell stetig verbessert werden.

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 an 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.

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. [↑](#footnote-ref-1)