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, integrierten 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 integriert, bereinigt 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 Modelle werden nicht nur über klassische Metriken wie Precision, Recall, F2-Score und R² bewertet, sondern zusätzlich anhand der realen wirtschaftlichen Wirkung. Auf Basis der Testdaten wird evaluiert, wie hoch die tatsächlichen Kontrollgewinne und -verluste wären, wenn man dem Modell folgt.

Aufgesetzt wird auf dem integrierten 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.

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 |
| 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 | Damit das Ergebnis von anderen nachvollziehbar ist, wurde ein fester random-state gewählt. |
| Skalierbarkeit | Das Modell sollte in allen Filialen des Kunden unabhängig von dessen Größe einsetzbar sein. |
| Robustheit | Die eingesetzten Verfahren sollten möglichst fehlerunanfällig sein. |
|  |  |

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

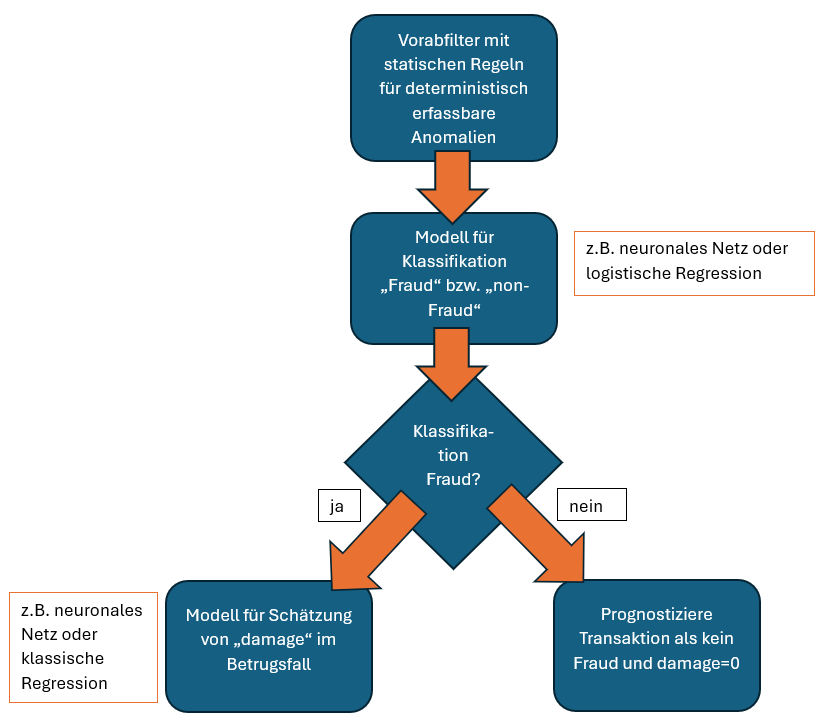
* **Kategorie „Übrige Rabatte“:** Diese Kategorie umfasst alle Rabatte, die in Produktkategorien auftreten, in denen reguläre Rabatte im Zusammenhang mit einem abgelaufenen MHD möglich sind. Allerdings ist auch in dieser Kategorie die Wahrscheinlichkeit für einen FRAUD signifikant höher als in den übrigen Datensätzen.
* **Kategorie „Übrige “:** Diese Kategorie umfasst alle übrigen Datensätze, die keiner der oben genannten Kategorien zugeordnet werden können.

NEU Raphael:

Wie bereits in Meilenstein 2 erläutert, 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

Vor dem Einsatz von Machine-Learning-Modellen wurden statische Regeln formuliert, die auf Domänenwissen und unseren oben dargestellten Ergebnissen der explorativen Datenanalyse beruhen. Sie dienen dazu, offensichtliche Betrugsfälle ohne Modellentscheidung direkt als FRAUD zu klassifizieren, um die Modellkomplexität zu verringern und die Präzision zu erhöhen.

Diese Regeln betreffen insbesondere:

* *Transaktionen, in denen Positionen vom Kamerasystem hinzugefügt wurden, weil ein Produkt gesichtet, aber nicht gescannt wurde* (has\_unscanned = True)
* *Transaktionen, in denen für Produktkategorien (z. B. Haushaltswaren) Rabatte aufgrund abgelaufener Mindesthaltbarkeitsdaten manuell angewendet wurden, obwohl solche Artikel typischerweise kein MHD aufweisen*

Diese Fälle werden vorab als FRAUD klassifiziert und aus der weiteren Modellbildung ausgeschlossen. Dadurch kann sich das statistische Modell auf die schweren zu bewertenden Fälle konzentrieren.

**Allerdings besteht ein Risiko, dass in einem anderen Umfeld als in unserem Trainingsdatensatz diese Regeln nicht zum Tragen kommen und das von uns eingesetzte Modell schlechtere Klassifikationsergebnisse liefert als geplant.**

**NEU Raphael:**

**Analyse einfacher statischer Entscheidungsregeln**

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.

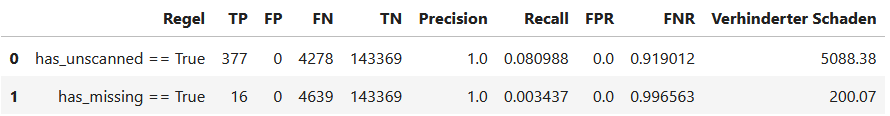
Zentrale Attribute bei der Regelbildung waren has\_unscanned und has\_missing. Es handelt sich um Transaktionen, bei denen von der Kamera fehlende Positionen erkannt wurden und automatisch in die Transaktion als FRAUD aufgenommen wurden (has\_unscanned) bzw. HIER NOCH EINE ERKLÄRUNG VON HASMISSING EINFÜGEN.

**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 dieser Stufe erfolgt die Vorhersage der **Wahrscheinlichkeit**, mit der eine Transaktion betrügerisch ist. Es wird zwischen zwei Modellklassen unterschieden, und zwar **Klassische** **Klassifikationsmodelle** einerseits sowie **Künstliche** **Neuronale** **Netze** andererseits.

1. **Klassische Klassifikationsmodelle**

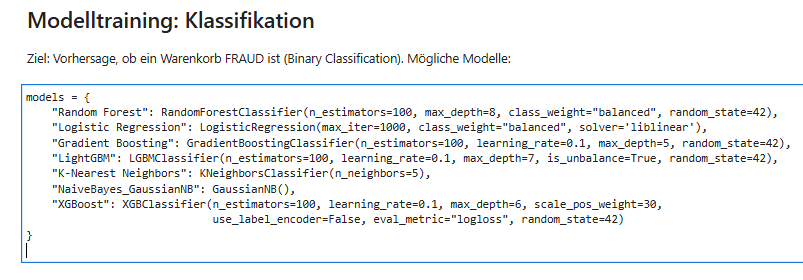
Die klassischen Modelle umfassen unter anderem Entscheidungsbäume, logistische Regression, Support Vector Machines, k-Nearest Neighbours sowie Ensemble-Methoden wie Random Forest und Gradient Boosting.

Diese Modelle liefern als Ergebnis eine geschätzte Wahrscheinlichkeit p für die Klasse „FRAUD“. In der klassischen Modellentwicklung würde ein Schwellenwert von 0.5 als Entscheidungskriterium verwendet. Aufgrund der stark unbalancierten Klassenverteilung ist dies jedoch nicht zielführend. Stattdessen dient das Modell in unserem Fall dazu, die Wahrscheinlichkeit für einen Betrugsfall zu schätzen, die in späteren Entscheidungsstufen weiterverwendet wird.

Ein Machine-Learning-Klassifikationsmodell wird trainiert, um die Wahrscheinlichkeit vorherzusagen, dass eine Transaktion als FRAUD zu bewerten ist. Dazu werden mehrere Modelle miteinander verglichen (RandomForest, Logistic Regression, LightGBM, XGBoost, SVM, KNN). Das beste Modell wird anhand von Präzision, Recall, F2-Score und der daraus resultierenden Entscheidungskonsequenz im Kostenmodell bestimmt.

**Vorgehen:**

* Aufteilung der Daten in Trainings- und Testmenge (80/20) unter Beibehaltung der Klassenverteilung (stratify=y\_clf)
* One-Hot-Encoding der kategorialen Variablen
* Vergleich mehrerer Modelle (Logistische Regression, Random Forest, Gradient Boosting)
* Auswahl des besten Modells basierend auf Precision, Recall und ROC-AUC

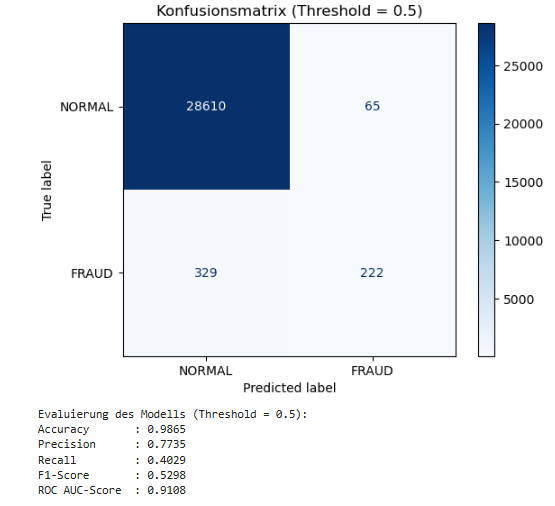


**Ergebnis des Modellvergleichs:**

Die folgende Übersicht zeigt die Ergebnisse unterschiedlicher Modelle auf Basis der Testdatensätze (29.226 Datensätze entspricht ca.20 Prozent der zur Verfügung gestellten Datensätze)

Als bestes Modell zur Vorhersage der Betrugswahrscheinlichkeit erwies sich der **Gradient Boosting Classifier**.

Die Konfusionsmatrix für dieses Modell ergibt sich aus der folgenden Abbildung:



1. **Künstliche Neuronale Netze**

Auch künstliche neuronale Netze wurden im Rahmen der Modellvergleiche getestet. Aufgrund ihrer grundsätzlich anderen Architektur (nicht-lineare Repräsentation, hohe Modellkomplexität) unterscheiden sie sich in Training, Interpretation und Anwendung deutlich von den klassischen Verfahren. Allerdings lag der Fokus dieses Projekts auf **erklärbaren** Modellen (vgl. mit **stabiler** Performance, weshalb Neuronale Netze nur ergänzend berücksichtigt wurden.

1. **Ergebnis**

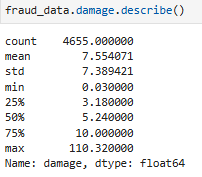
Ein Vergleich der beiden Modellklassen ergab keine signifikanten Unterschiede in der Leistungsfähigkeit. Aufgrund der oben definierten Anforderungen an Analyseverfahren liegt der Focus dieses Projekts auf **erklärbaren** Modellen mit **stabiler** Performance, weshalb Neuronale Netze nur ergänzend berücksichtigt wurden.

Für die weiteren Analysen wird auf dem besten klassischen Klassifikationsmodell „**Gradient Boosting Classifier**“ aufgebaut**.**

Stufe 3: Regressionsmodell zur Schadensschätzung

Ziel dieser Modellstufe ist die Schätzung der Schadenshöhe, die entsteht, wenn ein Betrugsfall nicht erkannt wird. Unabhängig davon, wie wahrscheinlich der Betrugsfall ist, liefert dieses Modell für jede Transaktion eine Schätzung der potenziellen finanziellen Auswirkung, unter der Annahme, dass es sich um einen tatsächlichen Betrug handelt.

Die Verteilung der Höhe der gesamten Schäden des Trainingsdatensatzes ergibt sich wie folgt:

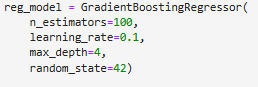


Da die Verteilung der Schadenshöhe (damage) stark rechtsschief ist, wurde auf die Verwendung eines linearen Regressionsmodells verzichtet. Stattdessen wurde ein Gradient Boosting Regressor zur Schätzung der Schadenshöhe eingesetzt. Diese Schätzwerte fließen in die Kosten-Nutzen-Abwägung der späteren Entscheidungsfunktion ein.

Für die als FRAUD klassifizierten Transaktionen sowie eine gleich große Anzahl als „NORMAL“ gelabelter Daten wird zusätzlich ein Regressionsmodell trainiert, um die zu erwartende Schadenshöhe (Spalte: damage) zu prognostizieren

**Modell:**

* **GradientBoostingRegressor**



**Ergebnis:**

* MAE (Mean Absolute Error): **ca. 2,04**
* R²: **ca. 0,38**
* Schadensschätzung stabil, auch für höhere Werte4

Stufe 4: Bewertung der Handlungsalternativen

In dieser Stufe wird auf Basis der beiden Modellprognosen – der Betrugswahrscheinlichkeit und der Schadensschätzung – eine wirtschaftliche Entscheidung getroffen: Soll eine Transaktion kontrolliert werden oder nicht?

**Entscheidungsstrategie und Kostenmodell**

Ziel ist es, auf Basis der Modellvorhersagen für jeden Warenkorb zu entscheiden, ob eine Kontrolle durchgeführt werden soll

Die Entscheidungsfindung erfolgt über den Vergleich der erwarteten Werte beider Alternativen, wobei sowohl Treffergewinne als auch Fehlalarmkosten und vermeidbare Schäden berücksichtigt werden. Mithilfe einer Bewertungsfunktion werden für jede Transaktion zwei Erwartungswerte berechnet:

**Alternative** 1: Kontrolle wird durchgeführt:

Der erwartete ökonomische Wert ergibt sich zu:

**Erwarteter Gewinn bei Kontrolle:**

*P(FRAUD) \** gain\_tp *− P(NORMAL) \* cost\_fp*

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

Der Erwartungswert berechnet sich wie folgt:

**Erwarteter Verlust bei Nicht-Kontrolle:**

*P(FRAUD) \* Erwarteter Schaden*

**Entscheidungsregel:**

Kontrollieren, wenn erwarteter Gewinn > erwarteter Verlust

Damit wird sichergestellt, dass nur dann kontrolliert wird, wenn es sich wirtschaftlich lohnt.

Schwellenwertoptimierung

Ziel ist es, auf Basis der Modellvorhersagen für jeden Warenkorb zu entscheiden, ob eine Kontrolle durchgeführt werden soll. Es wird diejenige Alternative gewählt, deren Erwartungswert höher ist. Diese Methode führt jedoch in der Praxis zu einer hohen Zahl von Fehlentscheidungen – insbesondere aufgrund der unausgeglichenen Verteilung zwischen FRAUD- und NORMAL-Fällen.

Um diesem Problem zu begegnen, wurden zwei zusätzliche Steuerungsmechanismen eingeführt:

**a) Schwellenwert für P(FRAUD)**

Nur Transaktionen mit einer geschätzten Betrugswahrscheinlichkeit oberhalb eines festgelegten Mindestwerts werden einer Kostenabwägung unterzogen. Dadurch werden sehr unsichere Fälle ausgeschlossen.

**b) Mindestdifferenz der Erwartungswerte**

Eine Kontrolle erfolgt nur dann, wenn der erwartete Nutzen der Kontrolle gegenüber der Nicht-Kontrolle eine definierte Mindestdifferenz überschreitet. Dies dient der Vermeidung von Kontrollen mit nur marginalem wirtschaftlichem Vorteil.

Für verschiedene Kombinationen beider Parameter wurde der Gesamtnutzen über den gesamten Testdatensatz berechnet. Die Parameterkombination mit dem geringsten Gesamtkostenwert wurde ausgewählt.

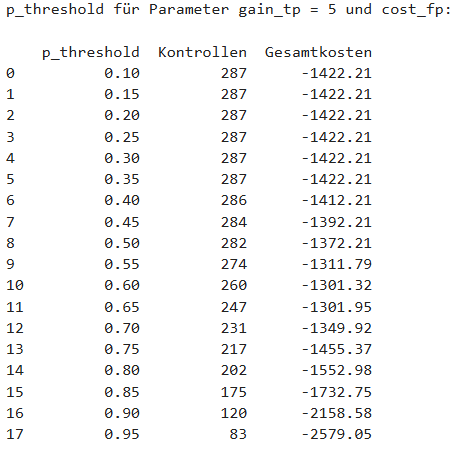
Wiederholung für verschiedene Modellvarianten

Die oben beschriebene Vorgehensweise (Klassifikation → Regression → Entscheidung) wurde für verschiedene Klassifikationsmodelle mit unterschiedlicher Parametrisierung wiederholt. Ziel war es, ein Modell mit optimaler Balance zwischen Präzision, Aufwand und Verständlichkeit zu identifizieren.

* Variation des minimalen Schwellenwerts für P(FRAUD) (z. B. von 0.1 bis 0.95)
* Entscheidung nur aktiv, wenn P(FRAUD) ≥ Schwelle
* Untersuchung des Effekts von gain\_tp (z. B. 3, 5,7,.. €) und cost\_fp (z. B. 2, 5, 10.. €)

Durch Variation des Schwellenwertes (z. B. ab welcher FRAUD-Wahrscheinlichkeit kontrolliert wird) und einer zusätzlichen Mindestdifferenz zwischen den Handlungsalternativen (z. B. 2 € Unterschied) wird die Strategie weiter verfeinert

Die folgende Tabelle zeigt die Gesamtkosten der Bewertungsfunktion sowie die Anzahl der durchgeführten Kontrollen auf dem Testdatensatz (29.226 Datensätze) in Abhängigkeit der Wahrscheinlichkeit für p(FRAUD):

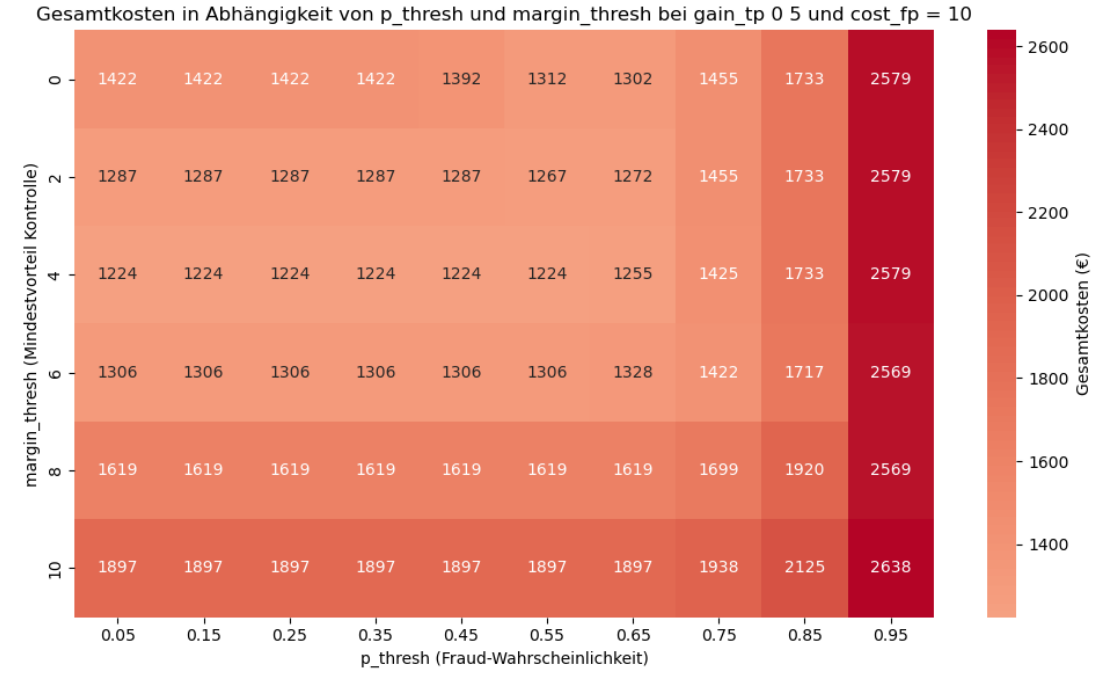


Zur Bewertung der Strategie wurden für jede Schwelle:

* Anzahl der Kontrollen
* Trefferquote (True Positives)
* Fehlalarme (False Positives)
* entgangene Schäden (False Negatives)
* gesamter betriebswirtschaftlicher Nutzen

berechnet.

Sofern im Modell zusätzlich ein margin\_thresh als Parameter berücksichtigt wird, so ergibt sich das folgende Ergebnis, wobei das Minimum der Gesamtkosten der Bewertungsfunktion bei 1.224,17 € liegt.

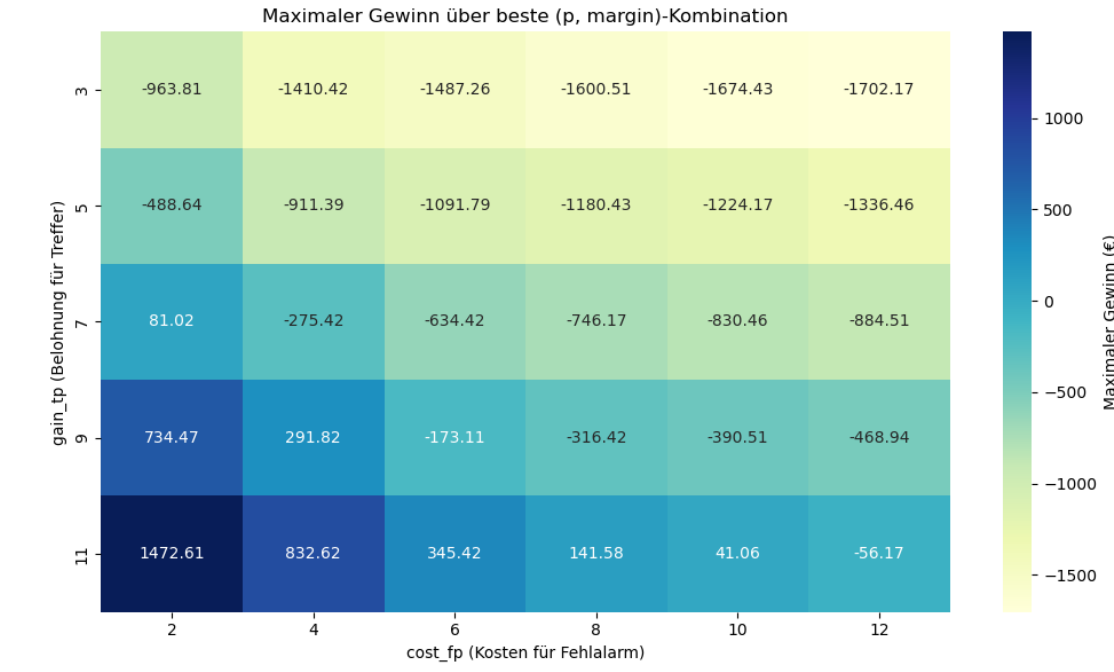


Sensitivitätsanalyse

Da sich diese Werte in der Praxis abhängig von Marktbedingungen, rechtlichen Rahmenbedingungen oder strategischen Überlegungen ändern können, wurde eine Sensitivitätsanalyse durchgeführt. Dabei wurde analysiert, wie sich Veränderungen dieser Parameter auf die Entscheidung und die Gesamtkosten auswirken.

Ein wesentliches Kriterium bei dieser Analyse ist, dass kein Modell schlechter abschneiden darf als die triviale Alternative, bei der gar keine Transaktionen kontrolliert werden. In diesem Fall entstehen Gesamtkosten in Höhe des gesamten, prognostizierten Schadens.

Aus diesem Grund wurde getestet, wie sich die optimalen Kosten in Abhängigkeit verschiedener Parameter (z. B. cost\_fp oder gain\_tp) verändern.



**Erkenntnisse:**

* Der wirtschaftlich optimale Schwellenwert liegt typischerweise bei 0.4–0.6
* Höhere cost\_fp-Werte führen zu restriktiveren Kontrollentscheidungen

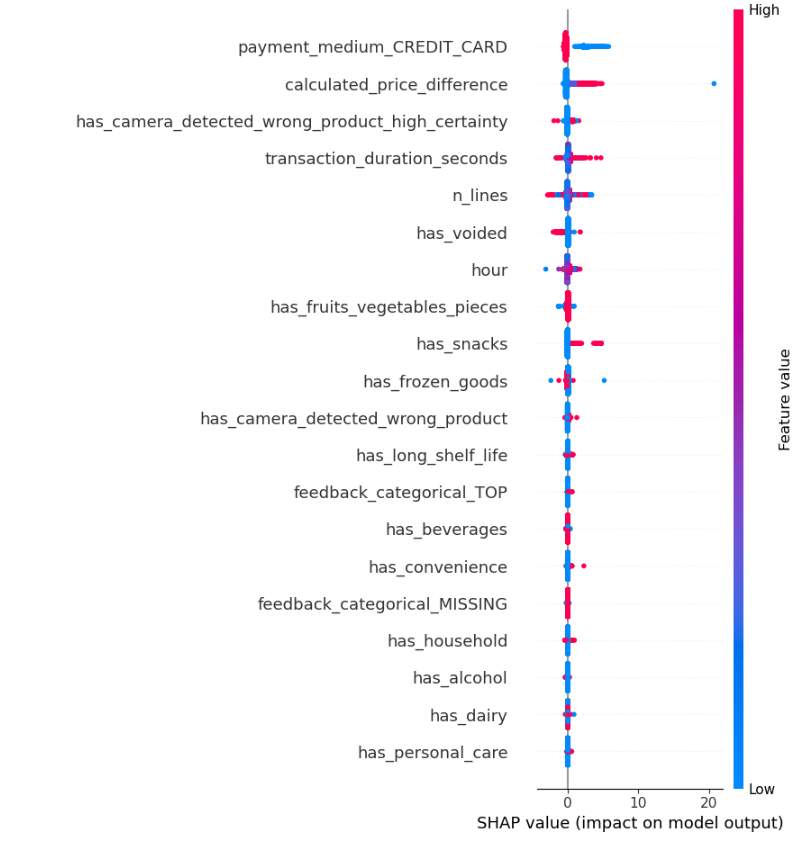
Eine Mindestdifferenz zwischen den beiden Erwartungswerten kann helfen, unsichere Entscheidungen zu vermeiden

Das vorgeschlagene Modell erreicht in allen untersuchten Szenarien mindestens dieselbe oder bessere ökonomische Performance als die triviale Nicht-Kontroll-Strategie. Die Gesamtkosten liegen in allen Testläufen unter dem prognostizierten Gesamtschaden.

Interpretation

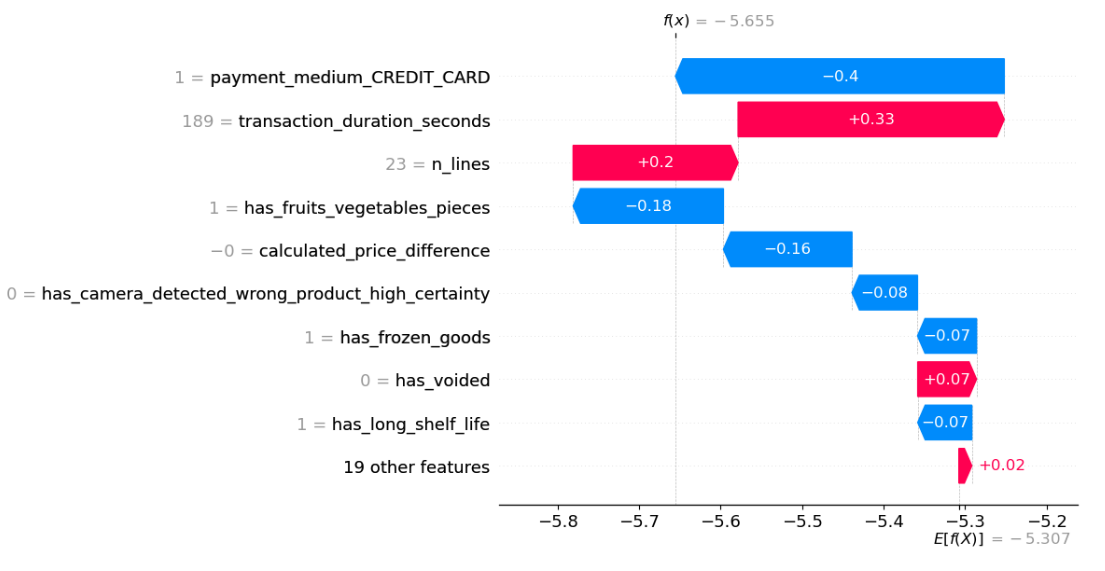
Für den Anwender des Modells stellt sich die Frage der Interpretation der Ergebnisse des Modells. Hierbei kann zwischen der allgemeinen Interpretation des Gesamtmodells einerseits und der Interpretation der jeweiligen Transaktion unterschieden werden.

Die folgende Abbildung liefert die allgemeine Interpretation des **gesamten** Modells und listet die relevanten und signifikanten Merkmale mit Einfluss auf die Modellentscheidung auf:

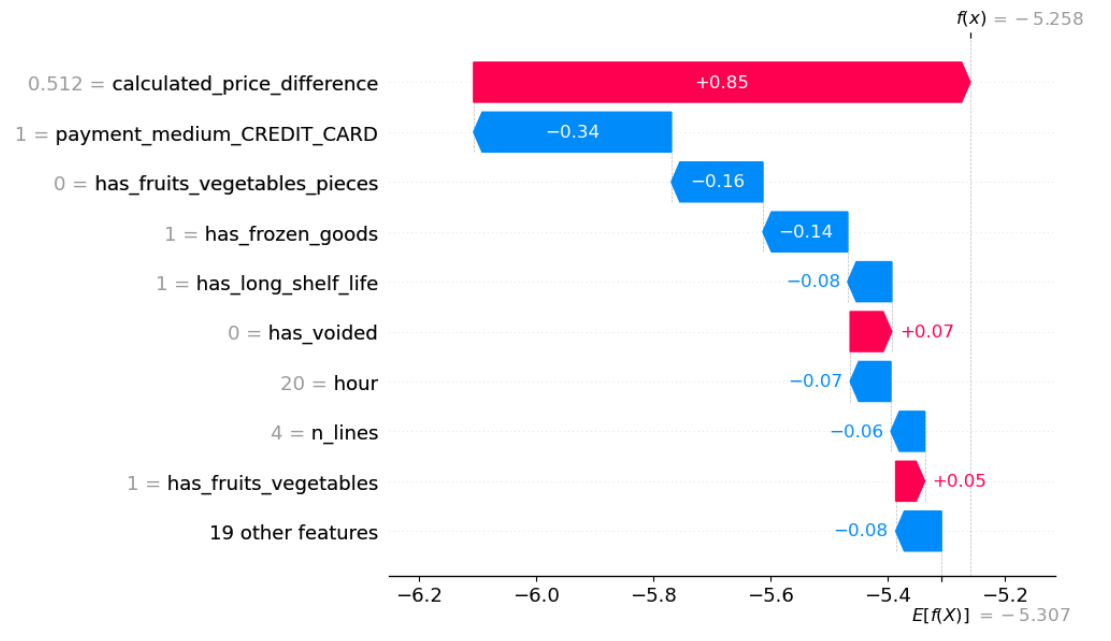


Die Interpretation zweier konkreter Transaktionen ergibt sich aus der folgenden Abbildung. Hier liefert der Wert E[f(x)] einen Wert für das Gesamtmodell, während der Wert f(x) den konkreten Wert der betreffenden Transaktion liefert. Die roten Kästchen bedeuten, dass die jeweilige Merkmalsausprägung einen positiven Einfluss auf die Wahrscheinlichkeit haben, während die blauen Kästchen das Betrugsrisiko vermindern.

**Beispiel** **1**: Transaktion mit **geringerer** Fraud-Wahrscheinlichkeit als im Durchschnitt



**Beispiel 2**: Transaktion mit **höherer** Fraud-Wahrscheinlichkeit als im Durchschnitt



Evaluation des gewählten Modells

Handlungsempfehlungen

Basierend auf den Analyseergebnissen werden folgende Maßnahmen empfohlen:

1. **Verbindliche Kontrolle bei erkannten Kamera-Hinweisen** unabhängig vom Modell
2. **Anwendung des Schwellenwertmodells** mit dynamischer Justierung der Kostenparameter
3. **Integration des Entscheidungssystems** in den operativen Ablauf über REST-Schnittstelle
4. **Regelmäßiges Re-Training der Modelle**, um Veränderungen im Kundenverhalten abzubilden
5. **Monitoring von FP/FN-Raten** im Echtbetrieb, um Modell-Drift frühzeitig zu erkennen

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
* Optional: Rückgabe von Entscheidungsgrundlagen (z. B. P(FRAUD), erwarteter Schaden)

**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.

Anhang

* Code-Auszüge (Modelltraining, Entscheidung, Schwellenanalyse)
* Grafiken zur Schwellenwertanalyse
* Datenkatalog der wichtigsten Features