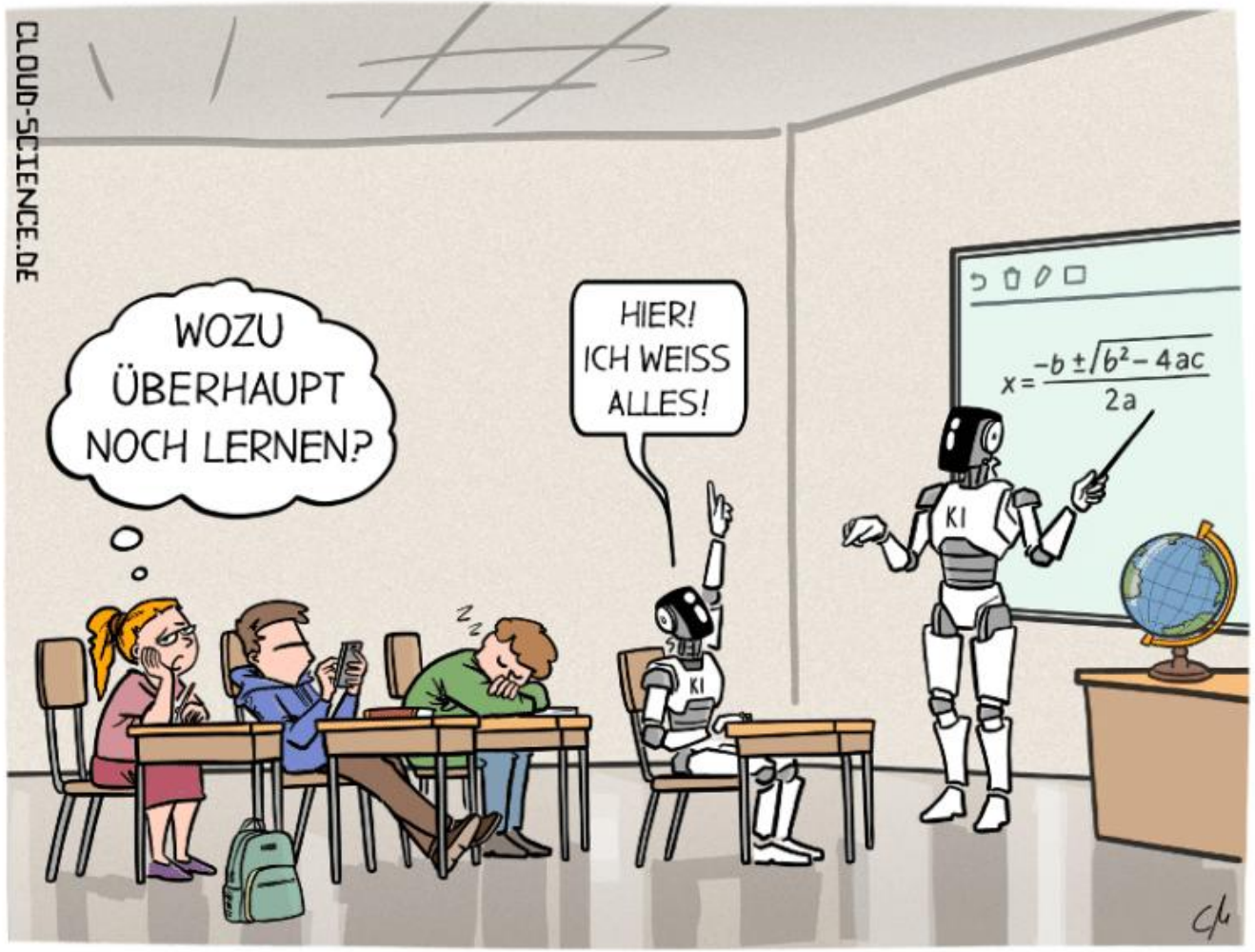
Abschlusspräsentation

Verlustprävention an Selbstbedienungskassen im Einzelhandel

David Zurschmitten
Matthias Bald
Raphael Schaffarczik



Themen für heute:

1. **Einleitung und Projektauftrag**
2. **Ergebnisse Meilenstein 2**
3. **Ergebnisse Meilenstein 3**
4. **REST-Schnittstelle**
5. **Abschlussbemerkungen**

1. Einleitung und Projektauftrag

Meilenstein1: Projektauftrag

- **Ziel des Projekts: Reduktion betriebswirtschaftlicher Schäden**
  - **Modell** zur Erkennung **auffälliger Muster/fehlerhafter Abläufe**
  - Algorithmus zur **Kennzeichnung verdächtiger Transaktionen**
  - **Konkreten Handlungsempfehlungen**

- **Nebenbedingungen**:
  - **Technische Machbarkeit**
  - **Echtzeitbetrieb & Skalierbarkeit**
  - **Betriebswirtschaftliche Sinnhaftigkeit** der Lösung
  - **Bewertungsfunktion** zur wirtschaftlichen Bewertung von Kontrollentscheidungen

Meilenstein 2: Datenzugang & Exploration

- **Zugang** zu Kassendaten (Transaktion, Artikel, Filiale)
- **Prüfung** auf Vollständigkeit, Struktur, Konsistenz
- Erste **explorative Analysen** & Hypothesenbildung
- **Evaluierung** der Modellierbarkeit (z. B. Labelverteilung, Datenqualität)
- Definition der **REST-Schnittstelle** für späteren Modellzugriff
- **Präsentation** erster Erkenntnisse & ggfs. Projektanpassung

Meilenstein 3: Datenanalyse & Modellierung

- **Datenbereinigung** & Erstellung eines Feature-Katalogs
- Auswahl geeigneter **Modellklassen** (von klassisch bis komplex)
- Integration der Bewertungsfunktion in die **Optimierungsstrategie**
- Erstellung eines funktionalen **Prototyps** zur Transaktionsbewertung
- **Bewertung** mit Kennzahlen (Precision, Recall, ökonomischer Nutzen)
- Ableitung konkreter **Handlungsempfehlungen**

Matthias Bald | David Zurschmitten | Raphael Schaffarczik

Meilenstein 4: Dokumentation & Übergabe

- Bereitstellung aller **Skripte, Modellartefakte & Visualisierungen**
- Übergabe des **Prototyps als Python-Paket**
- **Dokumentation der REST-Schnittstelle** zur einfachen Integration

Artefakte

Meilenstein 1:

1. abgestimmter Projektauftrag (PDF)

2. Präsentationsfolien über den Projektauftrag und die Projektstruktur (PowerPoint oder PDF)

Meilenstein 2:

3. Data Audit Report (fehlende Werte, Formatprobleme etc.)

4. Explorative Datenanalyse (erste Hypothesen über Datenmuster)

5. Präsentation der Erkenntnisse (Folien mit Visualisierungen)

6. Dokumentation zur geplanten REST-Schnittstelle

Meilenstein 3:

7. Feature-Katalog mit Beschreibung, Typ und Berechnungsmethode

8. Modellübersicht (getestete Modelle inkl. Parameter)

9. Bewertungsbericht (Precision, Recall, economic loss/gain etc.)

10. Visualisierung der Modelllogik

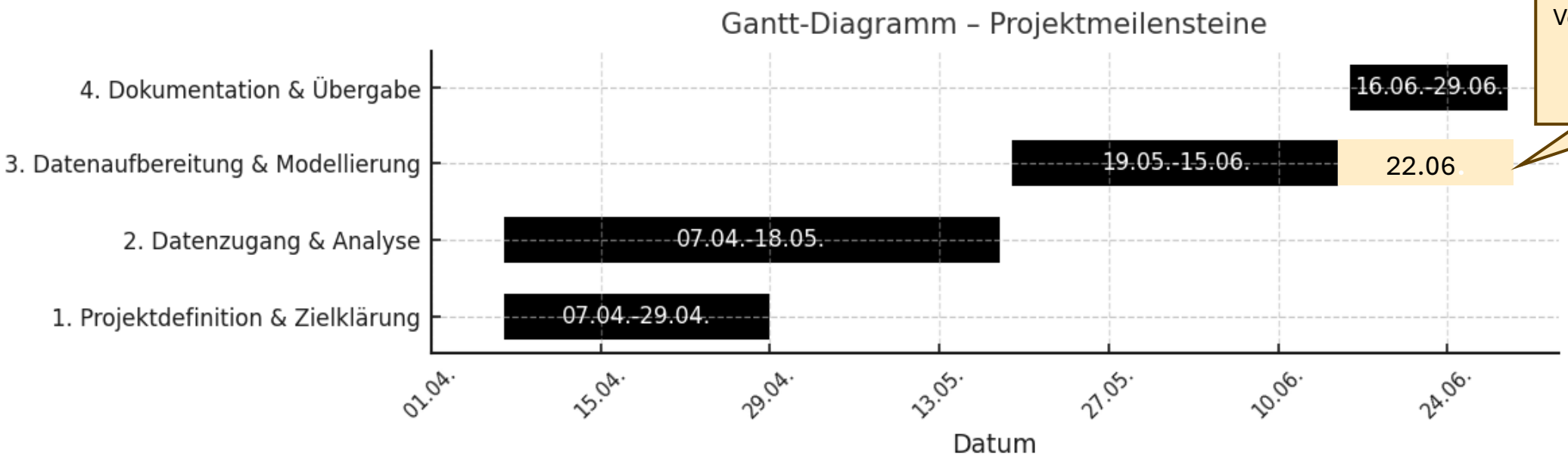
Meilenstein 4:

11. REST-Schnittstelle

12. Alle Programmskripte (Python)

13. Dokumentation

Meilensteinverlauf



Aufgrund verlegten Präsentationstermins Verlängerung um eine Woche

Risiken & Herausforderungen

- **Grenzen der Bewertungslogik** → Kleine Diebstähle könnten systematisch „übersehen“ werden
- **Überwiegender Teil nicht klassifizierter Daten** → Gefahr eingeschränkter Modellgeneralisation, v. a. bei komplexen Methoden
- **Übertragbarkeit auf andere Filialen fraglich** → Unterschiedliche Technik & Kundenverhalten
- **Nur Schadensfälle zu Lasten der Filialen** → auch „negative“ Schäden denkbar zu Lasten des Kunden

Abgrenzung des Projektumfangs

- **Keine Entwicklung** oder Empfehlung von:
  - **Hardware-Komponenten** (z. B. Gewichtssensoren, Kamerasysteme)
  - **Optischen Auswertungssystemen**
- **Keine juristische** Bewertung:
  - **Datenschutzfragen**
  - **Zulässigkeit** von Kontrollvorgängen
  - **Versicherungserstattungen**

Tools und Technologien

- **Programmierung in Python** (Jupyter Notebooks für Analysen, vollwertige Programme für abzuliefernde Schnittstelle)
- Codeversionierung und Dateiaustausch per **GitHub**
- Präsentationen und Dokumentationen in Microsoft **PowerPoint** bzw. Microsoft **Word (PDF)**
- Teammeetings per **Zoom** (ca. einmal pro Woche)
- Regelmäßiger Austausch per **Whatsapp**

Team & Aufgabenverteilung

Schwerpunkte:

Raphael (Data-Scientist, Mathematiker)

- Datenexploration
- Projektleitung

David (Softwareentwickler)

- Modellentwicklung und –vergleich
- REST-Schnittstelle

Matthias (Diplom-Kaufmann)

- Kommunikation mit Lehrstuhl
- Betriebswirtschaftlicher Teil und DASC-PM

3. Datenanalyse

Fokus des zweiten Meilensteins

- Fokus: Datenaufbereitung, Management & EDA (Explorative Datenanalyse)

- Vorbereitung für nachfolgende Modellierungsphasen: „**Exploration before prediction**“ – solide Basis für belastbare Modelle

- Teil des **iterativen Vorgehens** nach DASC-PM
  - Analysen & Modelle werden bei Bedarf angepasst
  - Neue Erkenntnisse oder zusätzliche Daten → Re-Validierung möglich

Übersicht der Datenquellen (1)

„products.csv“

- 8.120 Produkte aus 14 Kategorien
- Preis, Gewicht, Beliebtheit, Altersfreigabe …

„stores.csv“

- 18 Filialen mit Standortinfos, Bundesland, Urbanisierungsgrad

Karlsruhe	2
München	2
Köln	2
Bielefeld	1
Berlin	1
Leipzig	1
Kassel	1
Düsseldorf	1
Heidelberg	1
Bonn	1
Stuttgart	1
Chemnitz	1
Dortmund	1
Oberhausen	1
Osnabrück	1

Unsere Gruppe:

labeled:

Stuttgart	37921
Düsseldorf	37378
Köln	30061
Bonn	23110
Dortmund	19555

unlabeled:

Düsseldorf	377817
Stuttgart	377446
Köln	298486
Bonn	232882
Dortmund	195152

Übersicht der Datenquellen (2)

„transactions_train.parquet“

- 1.481.783 Transaktionen
- davon 148.025 gelabelte Transaktionen (NORMAL oder FRAUD)
- davon 4.656 mit erkanntem Betrug (FRAUD)
- Zeitstempel, Zahlungstyp, Kassennummer, Kundenfeedback…

„transactions_lines_train.parquet“

- 15.793.671 einzelne Kassenzeilen (Produkte) zu den Transaktionen
- Produkt-ID, Menge (Stück/Gewicht), Preis, Kamera-Sicherheitsklassifikation, Zeitstempel pro Scanvorgang…

Repräsentativität

Vergleich klassifizierter („gelabelter“, d.h. „FRAUD“ bzw. „NORMAL“) Daten mit dem restlichen Datensatz:

Numerische Spalten (t-Tests):

	Spalte	p-Wert	Mittelwert (labeled)	Mittelwert (unlabeled)	Std-Abw (labeled)	Std-Abw (unlabeled)
3	transaction_duration	0.185389	77.807475	77.541994	73.202614	72.895636
1	n_lines	0.355874	10.603607	10.575406	11.155176	11.101239
2	customer_feedback	0.671868	9.326005	9.318636	1.699571	1.715356
0	total_amount	0.750073	98.509750	98.413698	110.079582	109.943709

(Ausschnitt für numerische Merkmale auf Basis eines t-Tests)

Fazit: **gelabelte Daten sind repräsentativ** für den gesamten Trainingsdatensatz

Achtung: Unterschiede zwischen Trainings- und Testdaten!

Plausibilität

- Daten im Wesentlichen konsistent, aber:
  - **Komplexe Stornothematik** → konnte in Meilenstein 2 nicht abschließend geklärt werden, muss in Meilenstein 3 erneut aufgenommen werden
  - Durch statische Regeln lassen sich viele als „FRAUD“ deklarierte Transaktionen **sehr sicher vorhersagen**

- Berücksichtigung bei **späterer Modellbildung**

Transformation der Daten

- 4 Datentabellen in **eine einzige Datentabelle** überführt
- Relevante Transaktions- und Artikeldaten extrahiert bzw. berechnet
- Formatbereinigung und Überführung in analysierbare Tabellenstruktur
- **Pro Transaktion eine Zeile** erzeugt
- Artikelpositionen je Transaktion **zu Merkmalen aggregiert**

Aggregation der Daten

- **Positionsdaten zu Merkmalen aggregiert** (z.B. enthält Snacks, durchschnittliche Scanzeit pro Artikel etc.)

- Sowohl **kategoriale Merkmale** als auch **numerische**:
  - Tritt eine Kategorie in der Transaktion auf? Ja / nein
  - Wie viele Fälle? Anzahl

- **Transformation der Produktkategorien:**
  - Ist eine Produktkategorie vorhanden oder nicht (Getränke, Snacks, usw.)

- Minimum/Maximum/Mittelwert (Preis, Popularität, Zeit zwischen Scans)

Umgang mit unvollständigen Daten (1)

- Feedback: nur in **7,6 % der Fälle vorhanden**
  - Transformation zu kategorialen Ausprägungen (sehr gut, gut, mittel, schlecht, überhaupt vorhanden)

- 11.479 Fälle mit fehlenden Werten für mittlere und maximale Zeit zwischen Scans
  - Ursache: Nur ein Scan vorhanden
  - Ersetzt durch Mittelwert

Umgang mit unvollständigen Daten (2)

- 114 Fälle mit **fehlenden Werten des Kamerasystems**
  - Ersetzt durch den Modus

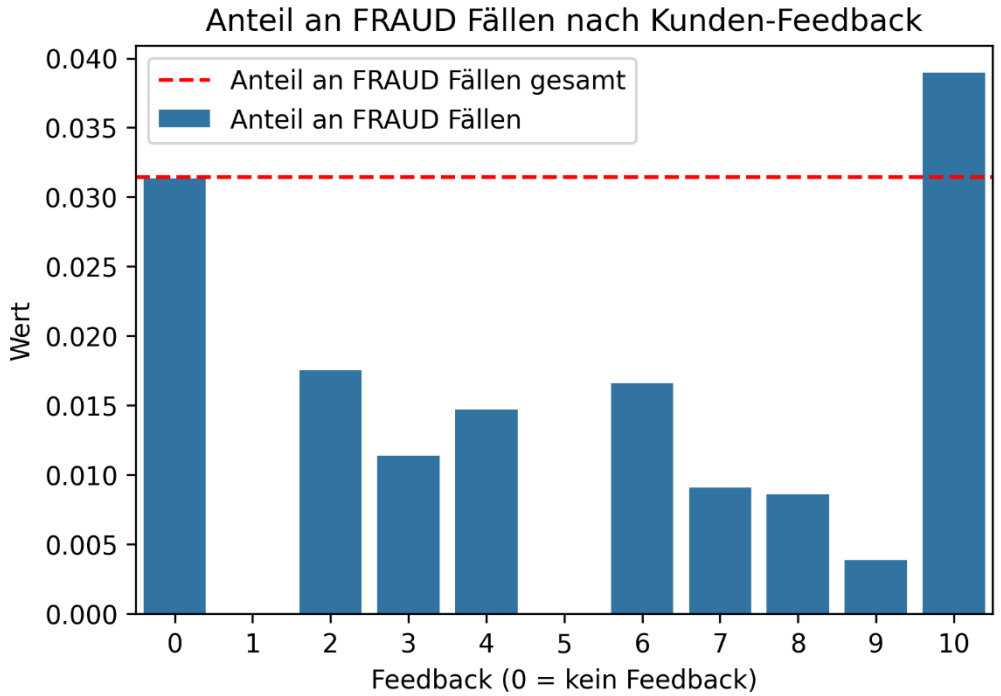
- Ein Fall mit mehreren fehlenden Spaltenwerten aufgrund fehlender Produkt-ID → entfernt

- Da wir nur die klassifizierten Daten betrachten → **keine Veränderung der nicht-klassifizierten Daten**

Übersicht

- 4 Schritte in der explorativen Datenanalyse:
  - **Verteilungsanalyse** und Ausreißer **numerischer** Attribute
  - Analyse **kategorialer Attribute**
  - **Nichtlineare Zusammenhänge** zwischen Attributen und Schadenshöhe
  - **Regressionsmodellierung**

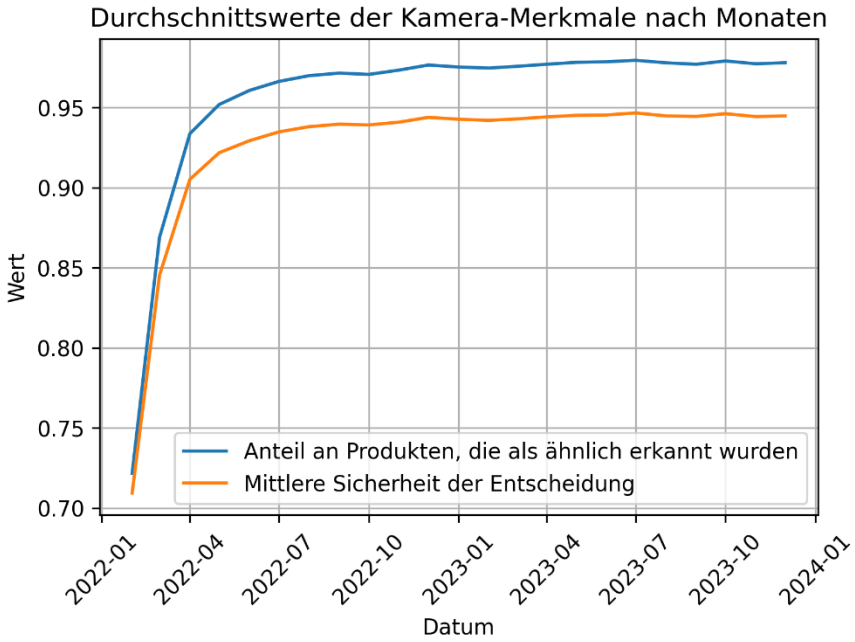
Auffälligkeit – Kundenfeedback



Wenige Werte bei Kundenfeedback und bei vorhandenen Werten extreme Ausprägung (bei Fraud mehrheitlich volle Punktzahl)

Lernkurve – Kamerasystem

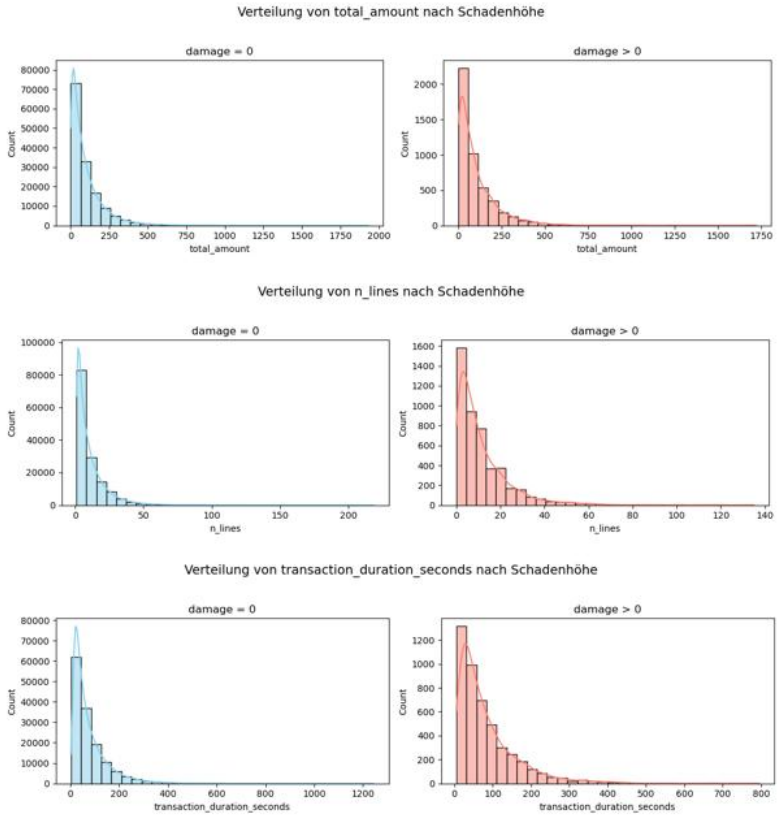
- **Kamerasystem anfangs nicht ausgelernt**
- Spätere Daten deutlich brauchbarer
- Zu beachten bei zukünftiger Einführung eines neuen Kamerasystems oder bei einer neuen Filiale



Numerische Merkmale von FRAUD (1)

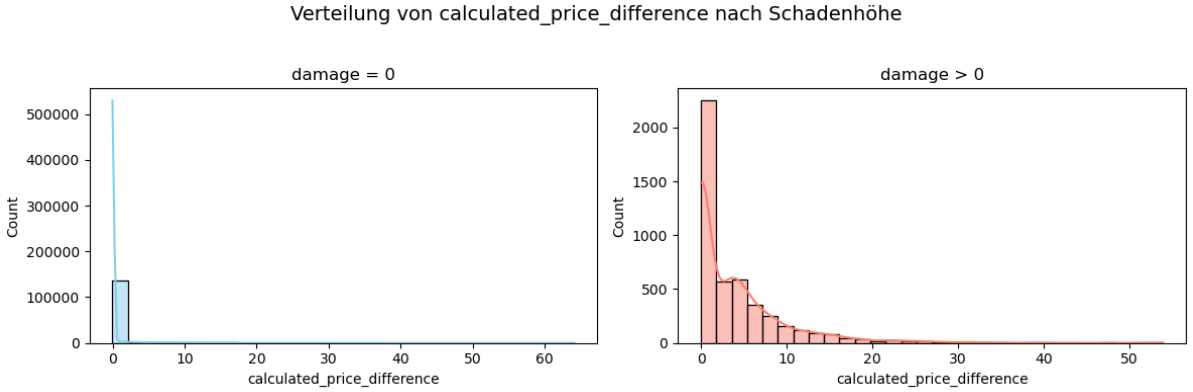
- Transaktionen mit Schaden (damage > 0):
  - **höhere Warenkorbsummen**
  - **mehr** gekaufte **Artikel** (n_lines)
  - **längere Transaktionsdauer**

- Merkmale sind **stark korreliert**
- **Interpretation:**
  - Mit wachsendem Warenkorb steigt die Komplexität
  - Fehler wie falsches Scannen oder vergessene Artikel werden wahrscheinlicher



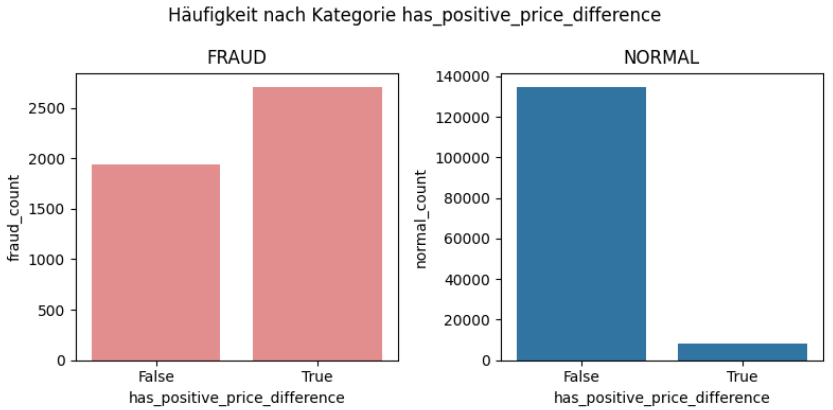
Numerische Merkmale von FRAUD (2)

- Transaktionen mit Schaden (damage > 0):
  - deutlich höhere *calculated_price_difference (*Differenz zwischen Summe der Einzelpreise und Kassensumme)
  - *calculated_price_difference* als potenziell **starker Prädiktor** für Verluste



Numerische Merkmale von FRAUD (3): Bezahlter Preis ≠ Nominalpreis

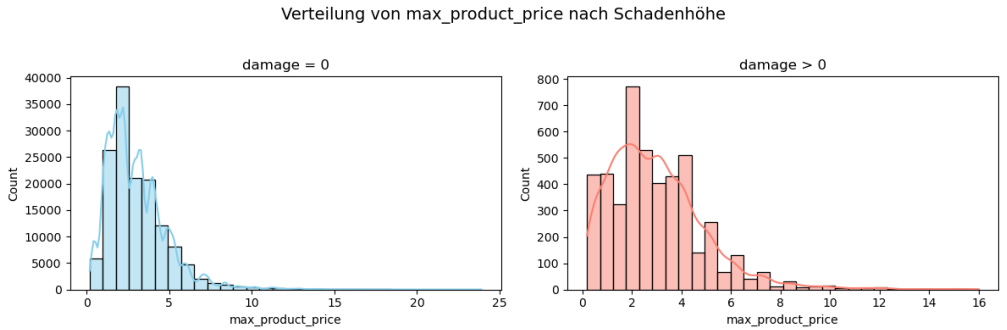
- **Nominalpreis** einer **Position**: Menge bzw. Gewicht multipliziert mit dem Nominalpreis des Artikels gemäß Produkttabelle
- **Nominalpreis** einer **Transaktion**: Summe der Nominalpreise aller nicht-stornierten Artikel
- **Häufige Abweichungen**
- Zwei definierte Merkmale:
  - **Differenz vorhanden** (ja/nein)
  - **Absolute Höhe** der Differenz

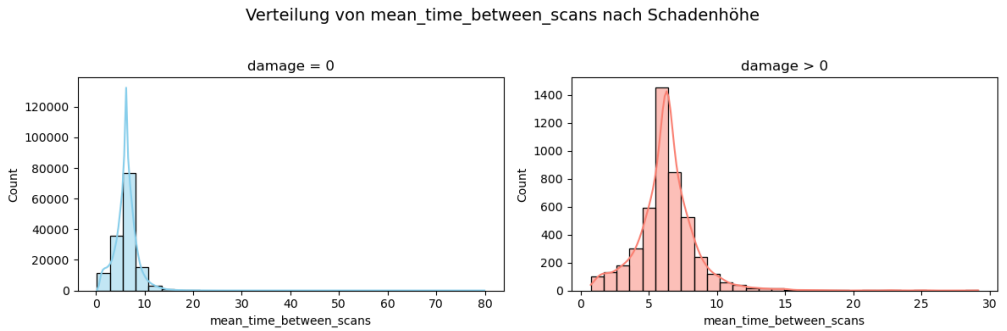


Matthias Bald | David Zurschmitten | Raphael Schaffarczik

Numerische Merkmale von FRAUD (4)

- Transaktionen mit Schaden (damage > 0):
  - enthalten leicht häufiger **hochpreisige Einzelartikel**
  - **breitere Streuung** bei der mittleren Zeit zwischen Scans





Numerische Merkmale: Extremwerte

- Für alle numerischen Features wurde der **Z-Score** berechnet
- Nutzen: Identifikation systematisch **auffälliger Attribute**
- Interpretation: Extremwerte nicht als Rauschen, sondern als **potenziell erklärungsstark** anzusehen
- Aber: Extremwerte **nicht systematisch häufiger** bei FRAUD als bei NORMAL

feature	outliers_abs_zscore>3
calculated_price_difference	3273
popularity_max	3193
total_amount	2962
ransaction_duration_seconds	2947
n_lines	2906
max_time_between_scans	2204
time_from_last_scan_to_end	2167
damage	2111
max_product_price	2073
mean_time_between_scans	1386
time_to_first_scan	949
popularity_min	161
days_since_sco_introduction	0

Numerische Merkmale: Signifikanz

- t-Test als Entscheidungskriterium, welche Prädiktoren signifikant sind
- Zusätzlich Analyse, wie viel mit dem Prädiktor erklärt werden kann (Relevanz)

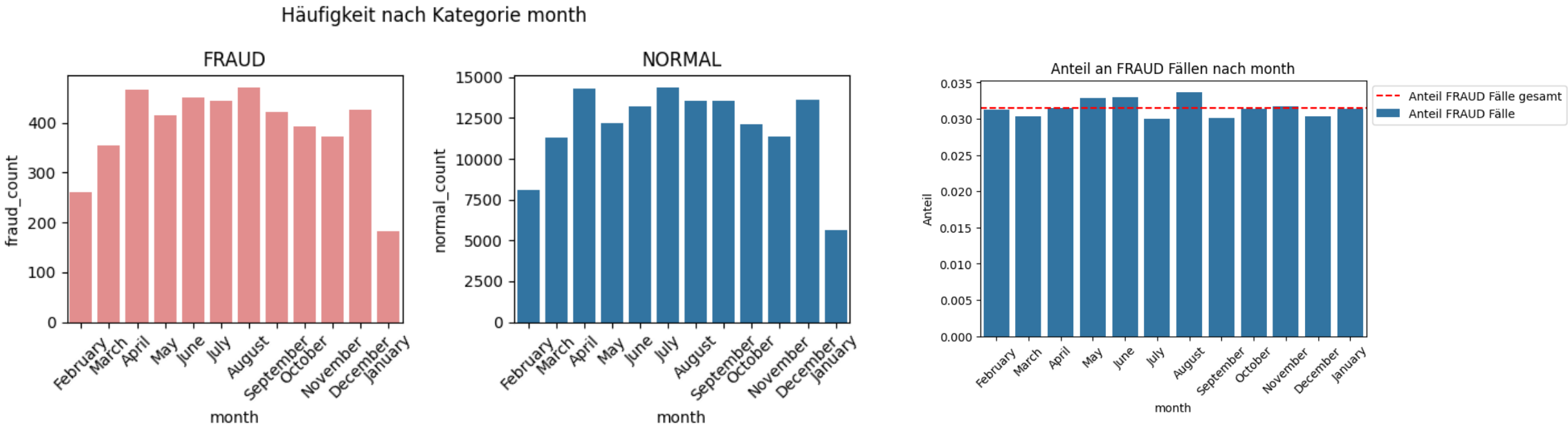
feature	significance	relevance
payment_medium	sehr signifikant	sehr relevant
hour	sehr signifikant	weniger relevant
[redacted]		
has_voided	sehr signifikant	weniger relevant
n_voided	sehr signifikant	weniger relevant
has_camera_detected_wrong_produc	sehr signifikant	weniger relevant
calculated_price_difference	sehr signifikant	sehr relevant
has_positive_price_difference	sehr signifikant	weniger relevant
has_snacks	sehr signifikant	weniger relevant

Kategoriale Merkmale von Fraud (1)

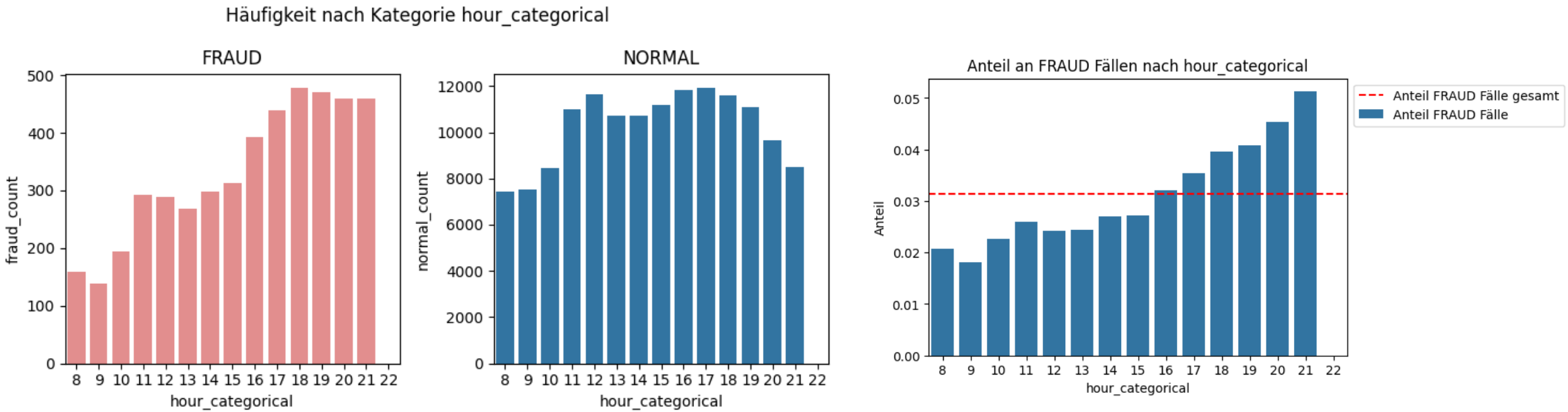
- Im Folgenden einige graphische Gegenüberstellungen von FRAUD / NORMAL anhand kategorialer Variablen

- Insbesondere bestimmte Produktkategorien kommen hier besonders häufig vor, ebenso:
  - Wurde mehrheitlich bar bezahlt
  - Hat das Kamerasystem Auffälligkeiten bemerkt

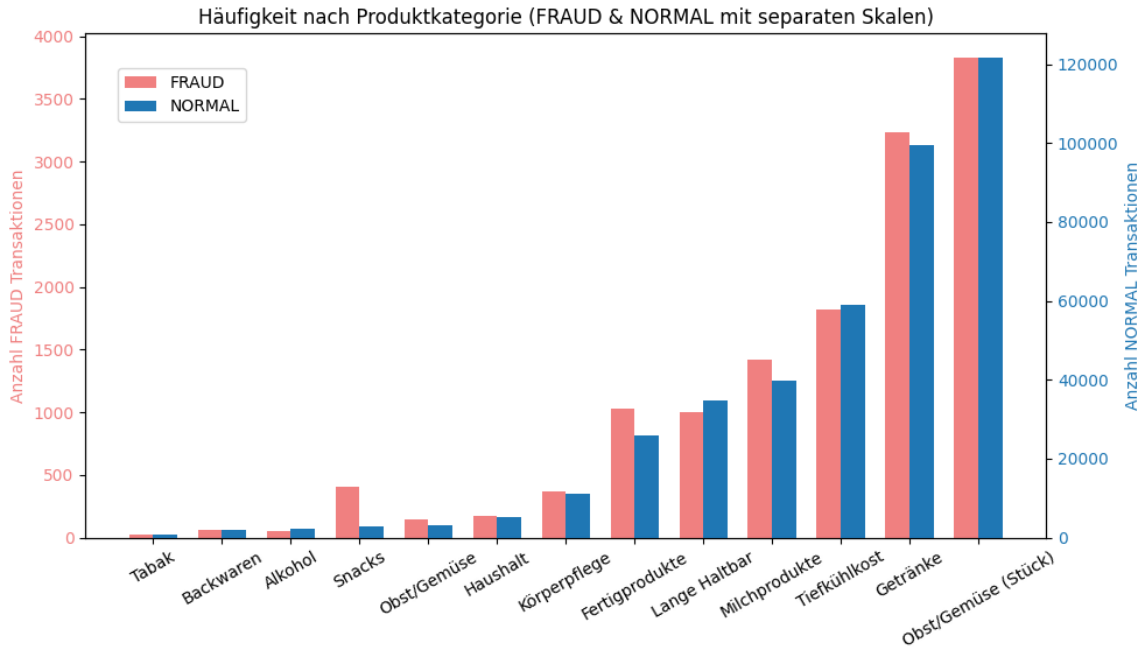
Kategoriale Merkmale: Monat

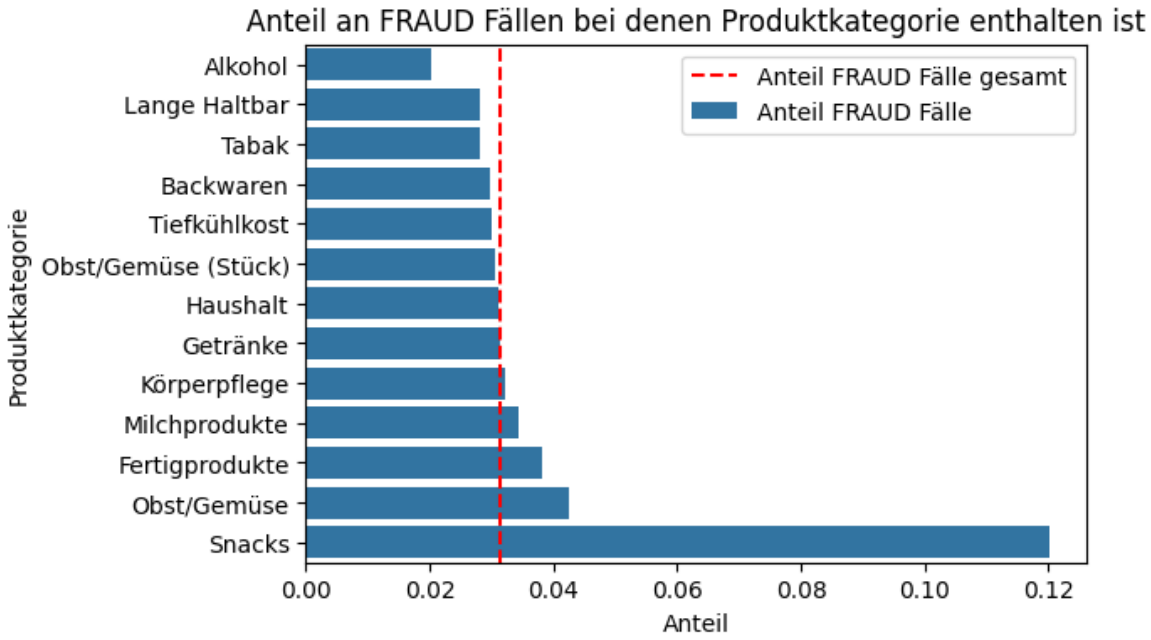


Kategoriale Merkmale: Tageszeit



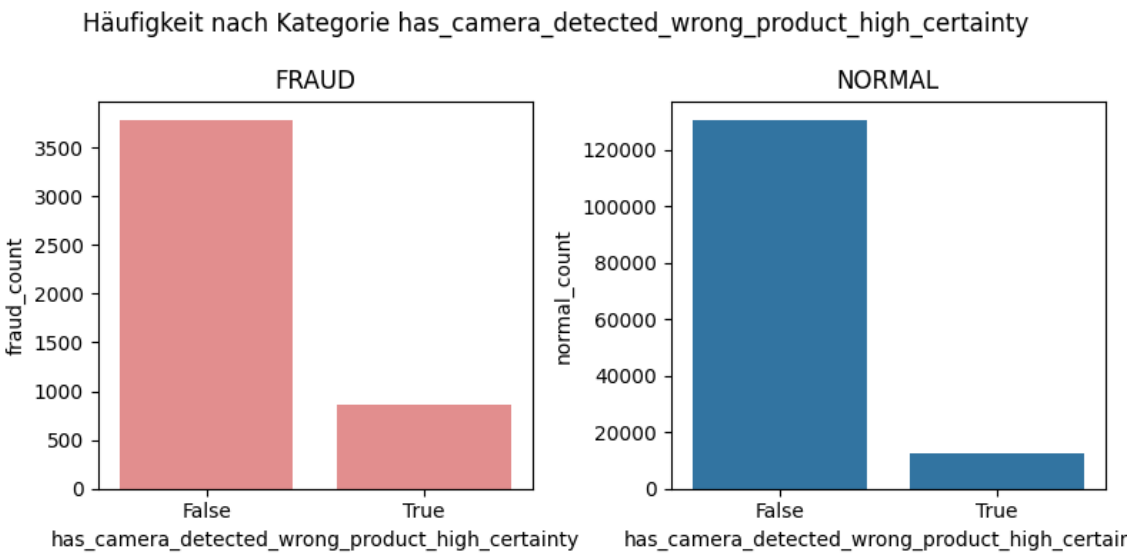
Kategoriale Merkmale: Produktkategorie

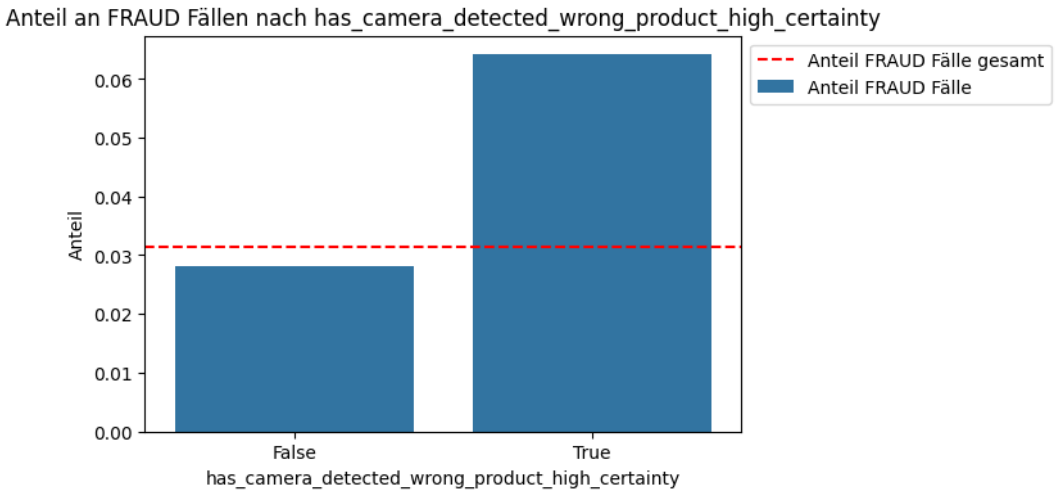




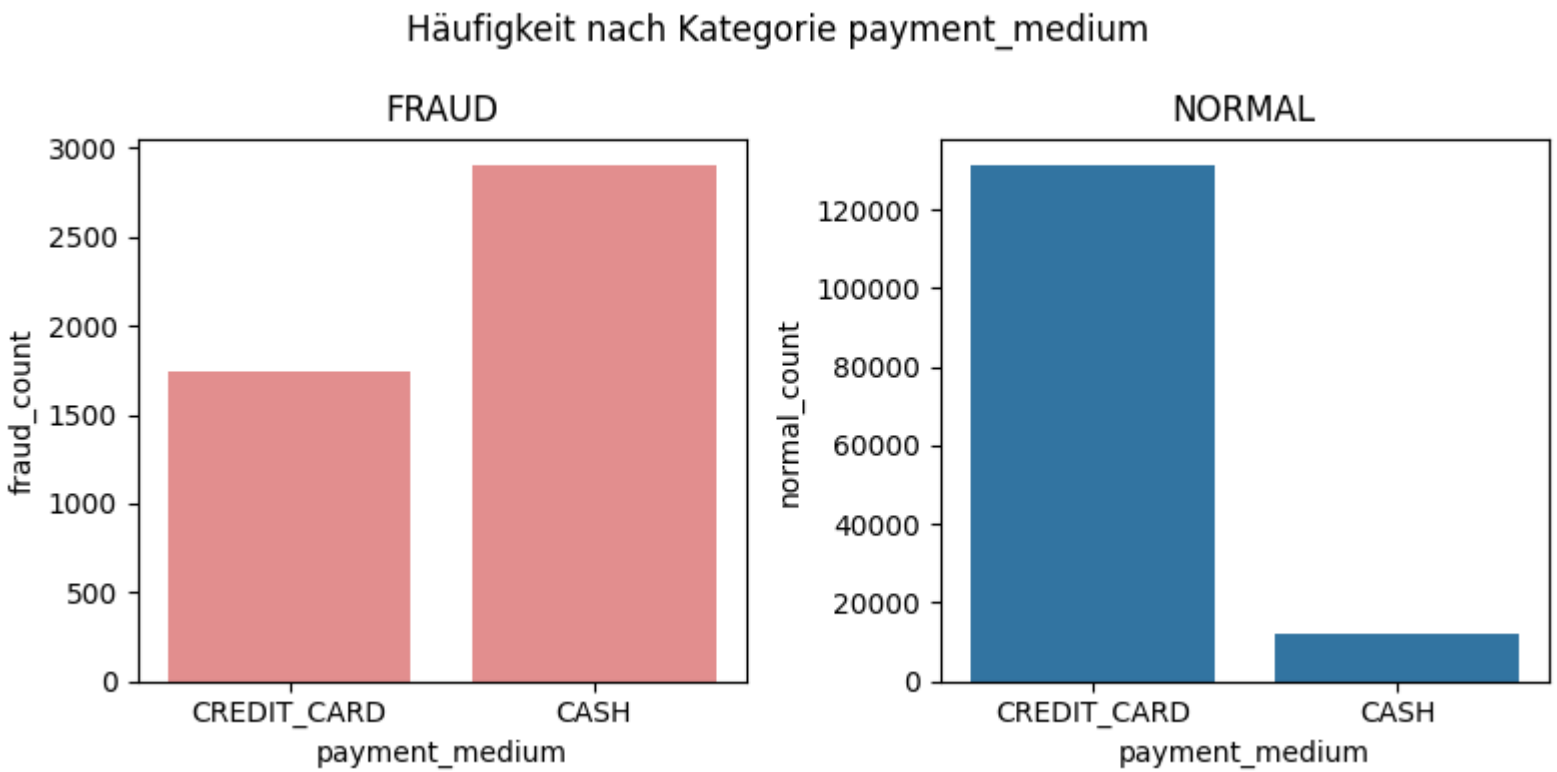
Matthias Bald | David Zurschmitten | Raphael Schaffarczik

Kategoriale Merkmale: Kamerasystem





Kategoriale Merkmale: Zahlungsmittel



Kategoriale Merkmale: Signifikanz

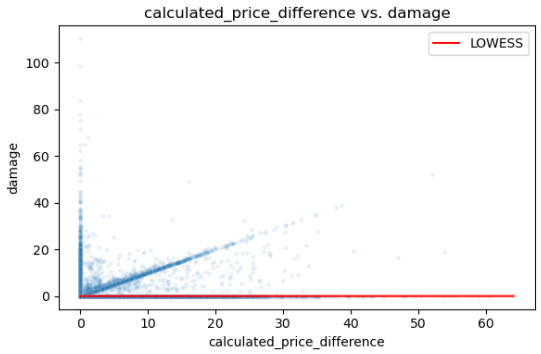
- Chi^2-Test als Entscheidungskriterium, welche Prädiktoren signifikant sind
- Zusätzlich Analyse, wie viel mit dem Prädiktor erklärt werden kann (Relevanz)

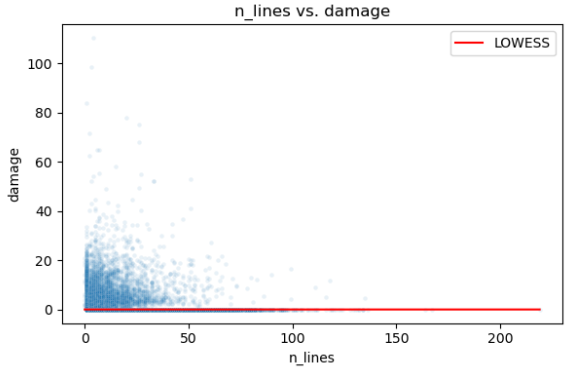
feature	significance	relevance
payment_medium	sehr signifikant	sehr relevant
calculated_price_difference	sehr signifikant	weniger relevant
has_positive_price_difference	sehr signifikant	sehr relevant

Nichtlineare Zusammenhänge

- Zur Analyse nichtlinearer Zusammenhänge zwischen numerischen Attributen und Schadenshöhe zwei Ansätze:
  - **LOWESS-Glättung** zur visuellen Trendbewertung
  - **Spearman & Pearson-Korrelation** zur quantitativen Bewertung

- Ergebnisse: Die meisten Merkmale zeigen keine klare nichtlineare Beziehung. Lediglich zwei Merkmale zeigen komplexere Beziehung zur Schadenshöhe.





Matthias Bald | David Zurschmitten | Raphael Schaffarczik

Regressionsanalyse: Multivariate Analyse

- **Multivariate Modellbildung** mit Reduktion (schrittweise Entfernen nicht relevanter Attribute)
- Getrennte Betrachtung für Zielgrößen:
  - Logistische Regression: FRAUD / NORMAL
  - Klassische Regression: Schadenshöhe

- Aufteilung der Daten in eine Trainingsmenge (80%) und eine Validierungsmenge (20%). Bewertung anhand der Performance auf beiden Mengen.

Regressionsanalyse: Auswertung

- Prognosegüte bei Klassifikation ist **verzerrt** durch die vielen Nicht-Schadensfälle; **bei ausgewogenem Datensatz bessere Performance**

- Geringe Vorhersagbarkeit der Schadenshöhe
  - **Breite Streuung** der Schadensbeträge
  - Großer Anteil an Null-Schäden → Verteilung verzerrt

- Komplexere Modelle mit Interaktionen:
  - **Verbesserung auf Trainingsdaten**, aber
  - **Kein Zugewinn auf Testdaten** → Überanpassung

```
Label-Modell:
Accuracy Test:  0.974
Accuracy Train: 0.974

Confusion Matrix Test:
Predicted    0.0  1.0
Actual
0.0        28646   43
1.0          712  204

Confusion Matrix Train:
Predicted     0.0  1.0
Actual
0.0        114498  182
1.0          2860  879
```

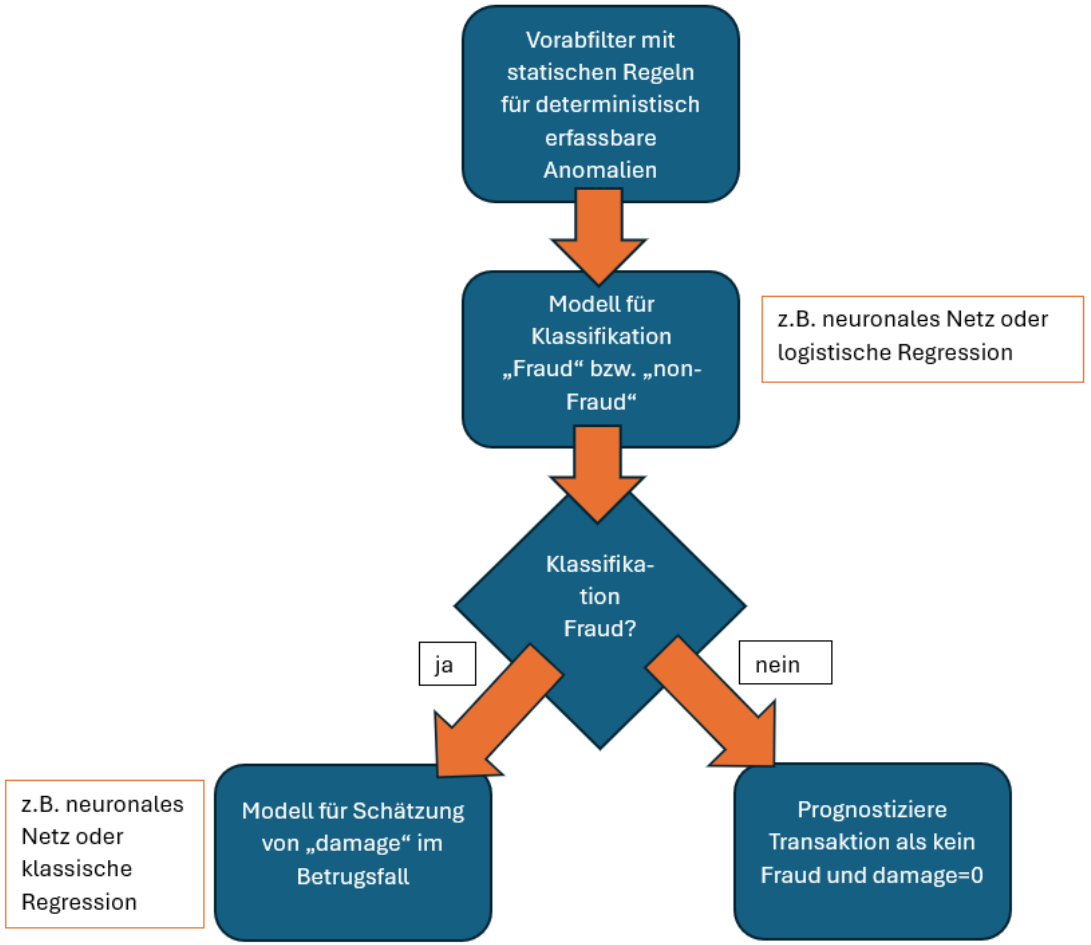
```
Damage-Modell:
R² Test:  0.137
R² Train: 0.136

RMSE Test:  1.754
RMSE Train: 1.721
```

Fazit des zweiten Meilensteins

- Daten sind **plausibel und konsistent** (Stornothematik noch zu klären)
- **Relevante Merkmale** wurden extrahiert und statistische analysiert
- Daten eignen sich für **weiteren Modellaufbau**
- **Komplexere Verfahren notwendig**, um durchgehend gute Prognosegüte sowohl bei der Klassifikation als auch der Schadensvorhersage gut abzuschneiden

Nächste Schritte

- **Dreistufiges Modell** auf Grundlage der aktuellen Datenerkenntnisse:
  - 1. Statische Anwendung gewisser **Erkennungsregeln**
  - 2. **Klassifikationsalgorithmus** zur Erkennung von fehlerhaften Transaktionen
  - 3. **Modell für Schätzung der Schadenshöhe** im Falle fehlerhafter Transaktionen (ansonsten prognostiziere Schaden=0)
  - Einbau der **Bewertungsfunktion** in Regeln für manuelle Kontrollen



4. Modellierung

Ziele des Meilensteins (1)

- **Ausgangsbasis**: bereinigte & aggregierte Transaktionsdaten
- **Ziel**: Entwicklung eines **praxistauglichen Modells** zur Betrugserkennung
- Berücksichtigung betriebswirtschaftlicher **Bewertungsfunktion**

	Tatsächlich FRAUD	Tatsächlich NORMAL
Vorhersage FRAUD	+5 (TP)	-10 (FP)
Vorhersage NORMAL	-Schaden (FN)	0 (TN)

- Werteverlust reduzieren <-> unnötige Kontrollen vermeiden

Ziele des Meilensteins (2)

Entwicklung eines mehrstufigen Modells:

- Einfache Regeln für offensichtliche Betrugsfälle
- Klassifikator für **Betrugswahrscheinlichkeit**
- Regressionsmodell zur **Schadenshöhe**
- Kombination beider Modelle in **Entscheidungslogik**

Konkrete Handlungsempfehlungen für den operativen Einsatz

Anforderungen an Analyseverfahren

- Mehr als nur Modellgüte: Entscheidungskriterien im Praxiseinsatz

- **weitere zentrale Anforderungen** gleichrangig berücksichtigt u.a.:
  - **Verständlichkeit:** Ergebnisse nachvollziehbar & visualisierbar
  - **Reproduzierbarkeit:** Konsistente Ergebnisse mit gleichem Code/Daten
  - **Umsetzbarkeit:** Einfach in der Praxis einsetzbar
  - **Skalierbarkeit:** Einsetzbar in allen Filialen, nachtrainierbar
  - **Robustheit:** Stabil bei Datenschwankungen & erneutem Training

Stufe 1: Statische Regeln zur Vorfilterung

- Ziel: einfache, **interpretierbare Regeln mit hoher Präzision bei minimaler Komplexität**

- **Methodik:**
  - Daten kategorial / binär kodiert
  - Analyse von Regeln mit ein bis zwei Merkmalen, um Überanpassung zu vermeiden und Interpretierbarkeit zu gewährleisten
  - Bewertung: Güte der Vorhersage höher als bei dem anschließenden Klassifikationsmodell

Kategorien von Fraud-Fällen

Kategorie	Anzahl Datensätze	NORMAL	FRAUD	Anteil FRAUD (%)	Gesamtschaden (€)
Kamera: ungescannte Artikel	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 €**

Rabattbetrug

- **Rabattsystematik auffällig**, aber schwer übertragbar auf neue Filialen → aktuell keine starren Rabattregeln implementiert.

- Mögliche technische Prävention:
  - **Rabattfunktion** bei nicht rabattfähigen Produkten **deaktivieren**
  - Nutzung vordefinierter **Rabatt-Barcodes** zur Kontrolle
  - Ergänzend: konfigurierbare **statische Modellregeln** (pro Filiale / Produktkategorie)

Statische Regeln zur Vorfilterung (2)

- Regeln mit einem Merkmal als Basis:
  - **has_unscanned** == True mit perfekter Vorhersage von Betrugsfällen
  - **has_missing** == True ebenfalls mit perfekter Vorhersage

- Wirtschaftlicher Nutzen **> 5.000 €** potenziell verhindert, aber nur **geringe Abdeckung** der gesamten Fälle (400)

	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 der statischen Regeln

- **Regeln mit zwei Merkmalen** enthalten entweder wiederum has_unscanned oder haben eine FPR von über 50% und sind daher **nicht sinnvoll;** nur Verwendung der beiden Einzelregeln
- Einzelregeln sehr präzise und ideal für vorgesehenen Einsatz
- has_unscanned & has_missing: FPR = 0, d. h. kein einziger False Positive Fall im Training
- **Ggf. Erweiterung** um zuvor besprochene statische Regeln gegen **Rabattbetrug**

Stufe 2a: Klassifikationsmodell

- Klassifikationsmodell liefert **Score zwischen 0 und 1** je Transaktion (→ Fraud- „Wahrscheinlichkeit“)
- Technisch keine echten Wahrscheinlichkeiten, aber gut interpretierbare Scores (nach Kalibrierung)
- Entscheidung erfolgt über einen **Threshold** (z. B. 0.5): Ab diesem Wert wird als FRAUD klassifiziert

Modellentwicklung & Evaluation

- **Iterativer Prozess** mit gezielter Auswahl leistungsfähiger Klassifikationsmodelle

- **4 zentrale Schritte**:
  - **Modellauswahl** & Vorabtests → ungeeignete Modelle ausgeschlossen
  - **Hyperparameter-Optimierung** & Kalibrierung der Scores
  - **Merkmalsauswahl** zur Reduktion & Robustheit
  - **Evaluation** mit Metriken & betriebswirtschaftlicher Bewertungsfunktion

Verglichene Modellklassen

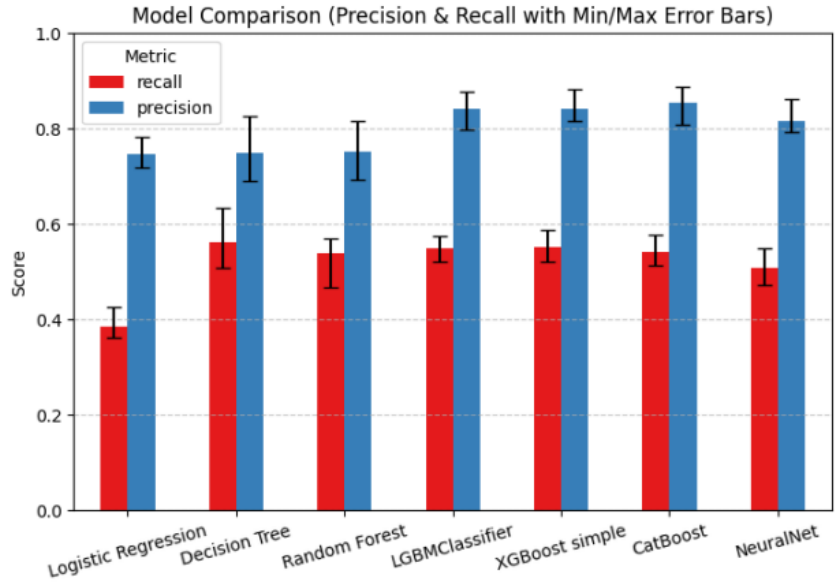
- **Baseline-Modelle:** Logistische Regression & Entscheidungsbaum
- **Fortgeschrittene Modelle:** Random Forest, Boosting (XGBoost, CatBoost), neuronale Netze
- **Boosting-Modelle** performten am besten → gezielte Weiterentwicklung
- **Neuronale Netze** zeigten gute Einzelresultate, aber instabil & sensitiv gegenüber Parametern

Modellvergleich (1)

- **Trainingsdaten ohne durch statische Regeln eindeutig erkannte Fälle**
- Lineares Modell: nur **3 ausgewählte Merkmale** aus Phase 2
- Alle anderen Modelle: **29 gezielt ausgewählte Features** (z. B. Zahlungsmittel, Tageszeit, Kamerasignale)
- Präprozessierung: One-Hot-Encoding + ggf. Skalierung
- Bewertung mit 5×5-facher Kreuzvalidierung unter Beibehaltung der Klassenverteilung (stratified CV)

Modellvergleich (2)

- **XGBoost, CatBoost, LightGBM** liefern beste Ergebnisse – sowohl statistisch als auch wirtschaftlich
  - **≈ 55 % der Fraud-Fälle erkannt**, hohe Präzision → wenige unnötige Kontrollen
  - +5 % Recall gegenüber neuronalen Netzen
  - **Logistische Regression deutlich schlechter** bei Recall & Schadenserkennung



	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

Modellvergleich (3)

- **XGBoost mit besserem Recall, CatBoost mit höherer Präzision** → Trade-off zwischen Entdeckungsrate und Kontrollkosten

- Random Forest unterliegt dem Einzelbaum – trotz Theorievorteil
  - Ursache: fehlende Hyperparameter-Optimierung

- CatBoost leicht besser, aber **Entscheidung zugunsten XGBoost** aus praktischen Gründen:
  - Starke Verbreitung
  - Gute Dokumentation
  - Effizientes Training
  - Besser wart- & erweiterbar im operativen Einsatz

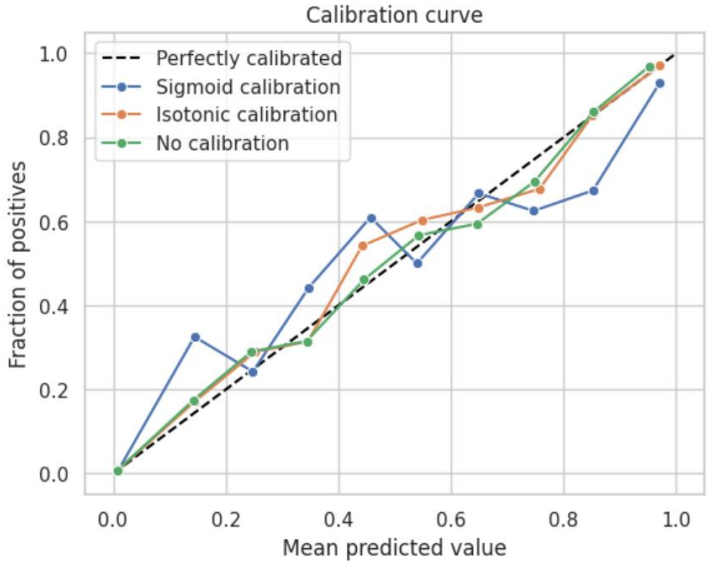
- **Erfüllt alle Anforderungen**: Verständlichkeit, Skalierbarkeit, Robustheit, Reproduzierbarkeit

Kalibrierung & Schwellenwertwahl

- XGBoost-Scores ≠ echte Wahrscheinlichkeiten, von daher sollte im Nachgang der Entscheidungsschwellwert (Sicherheit des Modells, dass Betrug vorliegt) kalibriert werden

- Aber: **Modell zeigt ohne Kalibrierung gute Performance**. Performance sogar besser als bei nachträglicher Schwellwertoptimierung!

- Standardwert 0.5 liefert stabilere & bessere Ergebnisse → keine Schwellenanpassung nötig



Stufe 2b: Regressionsmodell (Schaden)

- Ziel: **Schadenshöhe im Betrugsfall** prognostizieren – unabhängig von der Klassifikationssicherheit
- Ergänzt die Klassifikation um quantitative Risikoabschätzung pro Transaktion
- Ermöglicht differenzierte Entscheidungen: z. B. Kontrolle trotz niedriger Score-Wahrscheinlichkeit bei hohem vermutetem Schaden
- **Boosting-Modelle** erneut am besten, verwendet wird ein XGBoost-Regressor

Trainingsvarianten (1)

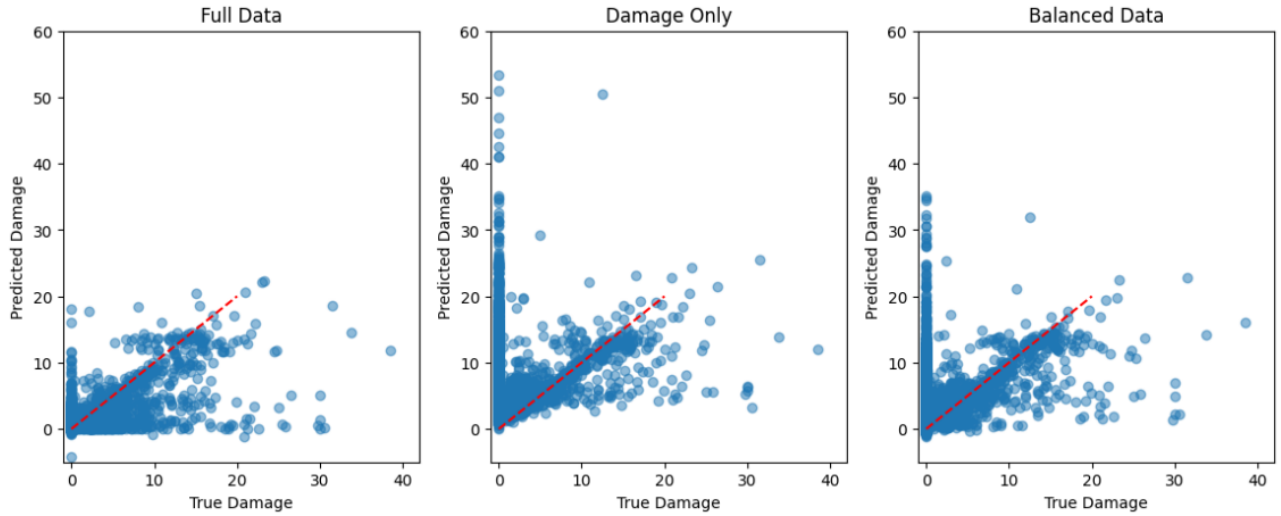
- 3 Trainingsvarianten zur Modellierung getestet:
  - **Vollständiger Datensatz** (inkl. Schaden = 0): realistisch, aber stark unausgewogen
  - **Balanced Set** (gleich viele Schaden / kein Schaden): sensitiv, aber nicht repräsentativ
  - **Nur Schadensfälle**: genauer für Betrug, aber nicht einsetzbar bei normalen Transaktionen

- Zielkonflikt: Generalisierung vs. Präzision vs. Repräsentativität

Trainingsvarianten (2)

- Alle Varianten haben Schwierigkeiten bei der Vorhersage hoher Schäden
- Im Bereich 0–10 €: hohe Streuung, quadratische Verteilung
- „Nur-FRAUD“-Modell überschätzt normale Transaktionen stark, trifft aber hohe Schäden am besten
- Klassische Metriken (R^2, MAE etc.) nur bedingt aussagekräftig im Vergleich
- „Damage-only“-Variante versagt bei Generalisierung, balanced liegt dazwischen

Resultate



Simulierte Bewertungsfunktion

- Ziel: wirtschaftlich sinnvolle Kontrollentscheidung für jede Transaktion

- Vergleich zweier Szenarien:
  - **Keine Kontrolle** → potenzieller Schaden bei nicht erkanntem Betrugsfall: P(FRAUD) * erwarteter Schaden
  - **Kontrolle** → Mischung aus erwarteter Fraud-Prämie (bei richtiger Klassifikation) & False-Positive-Kosten (bei Falschklassifikation): P(FRAUD) * 5 € – P(NORMAL) * 10 €

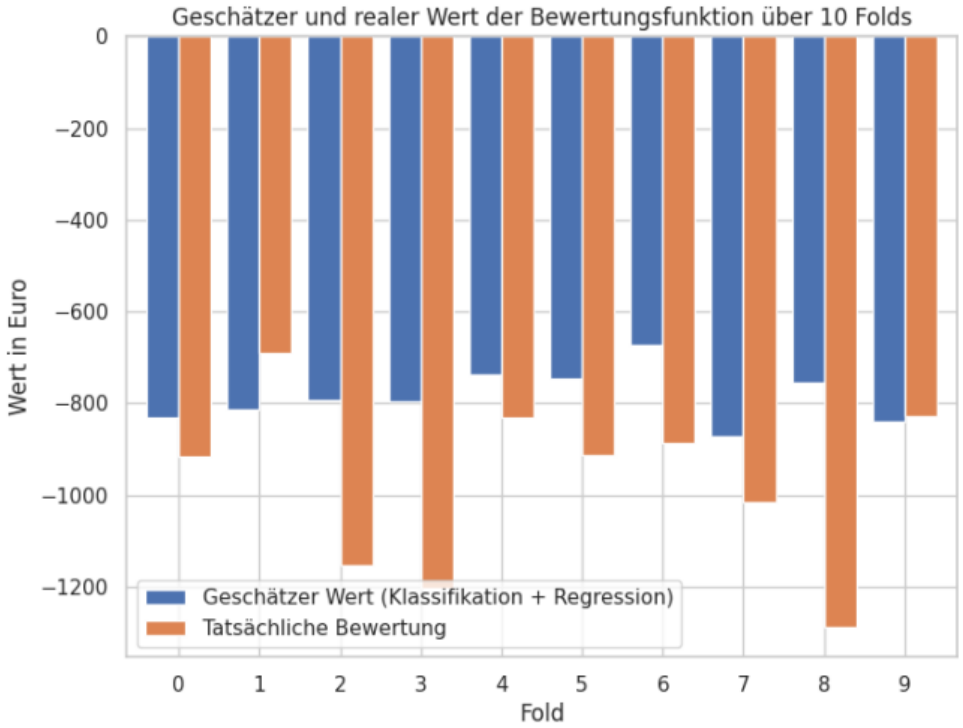
- **Kombiniertes Modell (Klassifikation + Regression)** simuliert Entscheidung für gesamten Datensatz
  - Wahrscheinlichkeiten P(FRAUD) bzw. P(NORMAL) aus Klassifikationsmodell
  - Erwartungswert des Schadens E(Schaden) aus Regressionsmodell

Simulierte Bewertungsfunktion

- Nur das Modell auf **vollständigem Datensatz** kann die tatsächliche Bewertungsfunktion realitätsnah approximieren

- Andere Varianten (balanced / damage-only) liefern massiv verzerrte Werte (falsche Mittelwerte: 5.1 / 4.7 statt 0.21)

- **Modell ist leicht optimistisch**, aber klar robustester Ansatz für praxisnahe Entscheidungen



Zusätzliche Optionen im Modell

- Neben den aktuell fixen Werten als **Belohnung bzw. Bestrafung** für richtig erkannte bzw. fälschlich als Betrug markierte Transaktionen können die Werte anhand der **Konfigurationsdatei** „beliebig**“ verändert** werden

- Die für **Rabatte ausgeschlossenen Produktkategorien** könnten ebenfalls per Konfiguration angepasst werden, sodass Rabatte in Kombination mit diesen Produktkategorien als Betrug (statisch) bewertet werden

Bilanz: Schaden durch Rabattbetrug

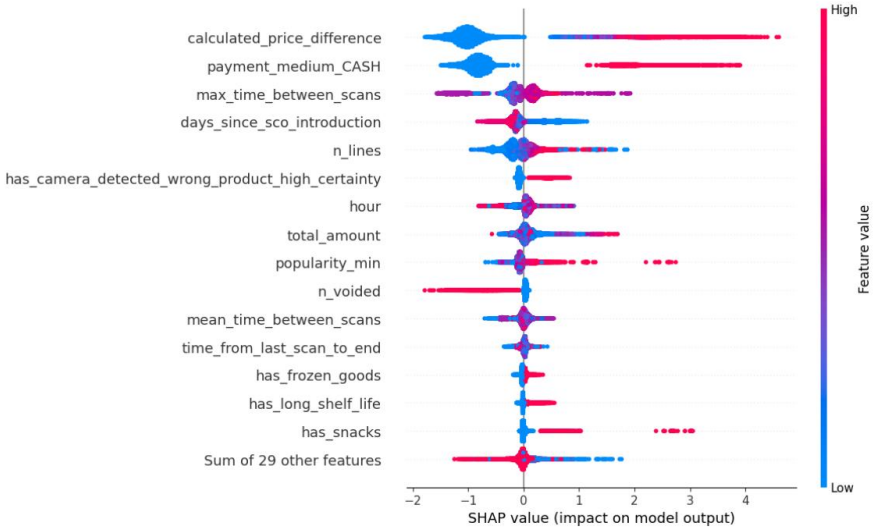
- Modell erkennt **62 % der unberechtigten Rabattfälle**
- Dadurch können im Schnitt 63 % des zugehörigen Schadens verhindert werden
- → **Hohes Präventionspotenzial** bei Rabattmissbrauch

Wirtschaftlicher Mehrwert des Modells

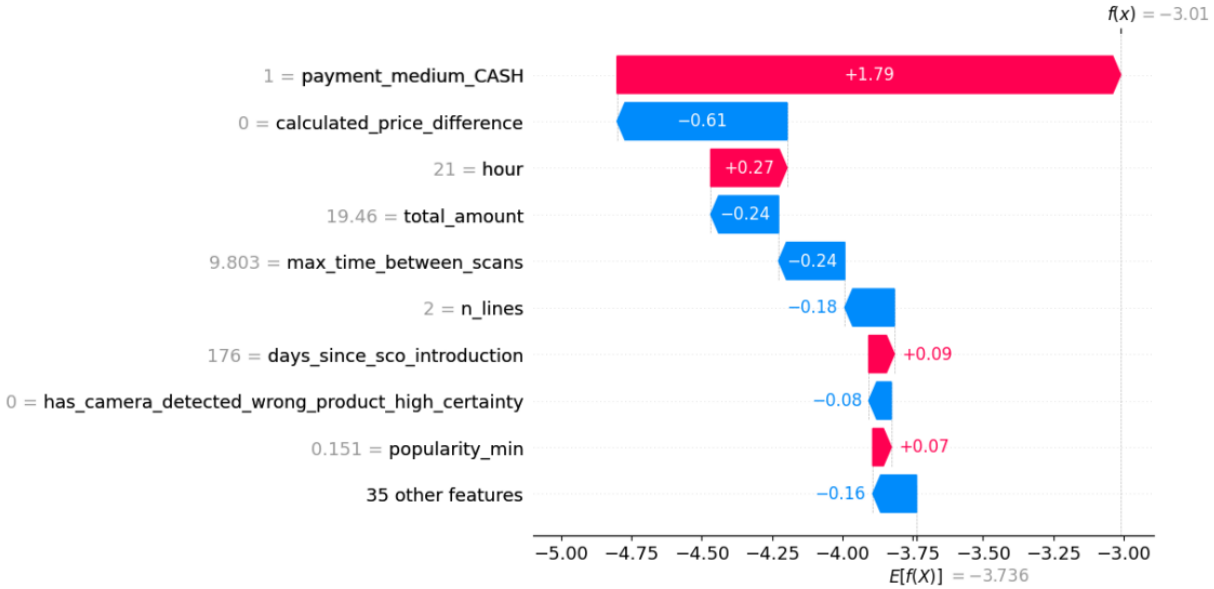
- Umfangreiche Evaluation: 200 Testläufe (5×CV mit 40 Wiederholungen)
- **Durchschnittlicher Mehrwert: 0,22 € pro Transaktion** (nach Bewertungsfunktion)
- Betrachtet man nur den verhinderten Schaden, also ohne Abzug der Kontrollkosten und ohne Bonus für entdeckte FRAUD-Fälle, ergibt sich ein mittlerer Wert von **0,15 € pro Transaktion**.
- Ergebnis gilt als robuste, belastbare Schätzung der Modellleistung
- Bezieht sich auf Testdaten (20 % des Gesamtbestands)

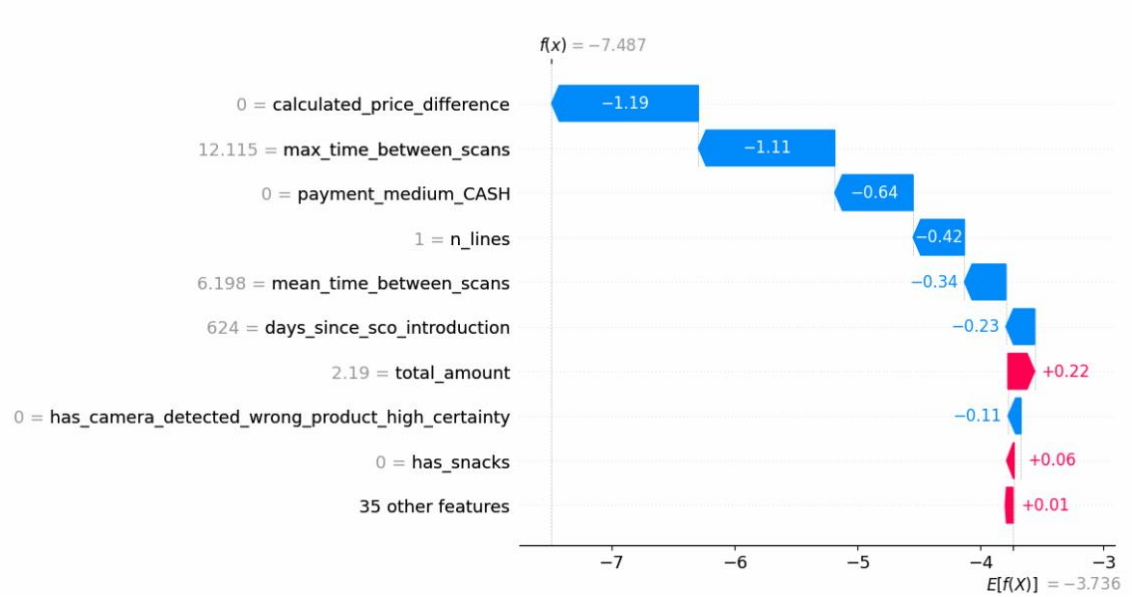
Sensitivitätsanalyse: Einflussfaktoren im Modell (1)

- Wichtigste Prädiktoren:
  - **Bargeldzahlung** = stärkster Einzelindikator für Fraud
  - **Preisabweichung & Kamera-Hinweise** erhöhen Risiko deutlich
- Zeitliche Merkmale (Scan-Dauer, Tageszeit) mit moderatem Einfluss
- **Jüngere Kamerasysteme** liefern **weniger aussagekräftige Daten**
- Modell trifft **nachvollziehbare Entscheidungen**, keine Blackbox



Sensitivitätsanalyse: Einflussfaktoren im Modell (2)





Matthias Bald | David Zurschmitten | Raphael Schaffarczik

Handlungsempfehlungen & Modellpflege

- **Gesamtmodell** (Statische Regeln + Klassifikation + Regression) ist betriebsreif → **zufällige** Kontrollen durch **datenbasierte Entscheidungen** ersetzen

- Ergebnis: Schäden vermeiden & Personal effizienter einsetzen

- Regelmäßige **Rekalibrierung** empfohlen bei:
  - neuen Filialen / Sortimenten
  - veränderten Kundengruppen
  - neuen Kameradaten / Kontrollrückmeldungen

5. REST-Schnittstelle

Technische Umsetzung

- Modell wird über eine **REST-API** ins Kassensystem integriert
  - Input: Transaktionsdaten (JSON)
  - Output: Echtzeitentscheidung + Schadenprognose + Begründung

- **Codeversionierung & Nachtraining** über GitHub möglich

- **Evaluation** mit Echtdaten der Wertkauf GmbH geplant

- Langfristige **Erweiterung denkbar**: z. B. durch Kundenhistorie, Treuekarten, Warenkorbinhalte

Schnittstelle im Detail

- Die in der Schnittstelle genutzten Modelle sind die vortrainierten besten Modelle der Trainingsdaten. Es findet **kein neues Training** innerhalb der Schnittstelle statt.

- Per **Konfigurationsdatei** können beliebige Werte für die Bewertungsfunktion verwendet werden (anstelle der +5 bzw. -10)

- Eingehende Daten (auf Positions- und Transaktionsebene) müssen entsprechend **zusammengeführt und aggregiert** werden (nur eine Zeile pro Einkauf)

- Entscheidungen des Modells sind nur teilweise auf prägnante Merkmale zurückzuführen. Deshalb keine klare Begründung, warum eine Transaktion verdächtig ist. Lediglich **Nennung der Parameter Schadensschätzung und FRAUD-Wahrscheinlichkeit**.

6. Abschlussbemerkungen

Zusammenfassung Modell (1)

- Klassifikation aller Transaktionen mit **has_unscanned = TRUE oder has_missing = TRUE als FRAUD** (100% TPR und knapp 400 Fällen in der Abdeckung auf den Trainingsdaten)

- Restliche Daten gehen in das **Klassifikationsmodell (unkalibriertes XGBoost-Modell)**

und in das

- Modell für die Vorhersage der **Schadenshöhe** (**XGBoost-Regressionsmodell** trainiert auf allen Trainingsdaten)

- Schnittstelle: Echtzeitentscheidung für oder gegen eine Kontrolle mit Begründung und Schadenprognose
    - **Keine Kontrolle** → potenzieller Schaden bei nicht erkanntem Betrugsfall: P(FRAUD) * erwarteter Schaden
    - **Kontrolle** → Mischung aus erwarteter Fraud-Prämie (bei richtiger Klassifikation) & False-Positive-Kosten (bei Falschklassifikation): P(FRAUD) * 5 € – P(NORMAL) * 10 €

Zusammenfassung Modell (2)

- **Aufgeteiltes Modell** (Trennung von statischen Regeln, Regressions- und Klassifikationsmodell) ist:
  - ökonomisch **nachvollziehbar**
  - Zeigt **solide Prognosegüte**
  - Lässt **Echtzeitentscheidungen** zu
  - Lässt sich **flexibel** mit anderen Straf- und Belohnungstermen (als die aktuelle Bewertungsfunktion) **konfigurieren**, ohne erneut trainiert werden zu müssen

- Zusätzlich sind alle einzelnen **Module** weitestgehend **unabhängig voneinander** und auch isoliert weiter optimierbar

Vielen Dank für Ihre Aufmerksamkeit!

Fragen & Anregungen?