Fakultät für
Mathematik und Informatik

Erkennung fehlerhafter Bezahlvorgänge an Selbstbedienungskassen im Einzelhandel

Abschlusspräsentation

Dominik Lewin, Mario Teßmann, Johannes Winkler

04.07.2024

FernUniversität in Hagen

Agenda

- **Projektauftrag**
- **Datenbereitstellung**
  - Datenvorverarbeitung
  - explorative Datenanalyse
- **Analyse**
  - Vorgehensweise
  - Details zum endgültigen Modell
  - Mehrwert des Klassifikators
- **Nutzbarmachung**
- **Zusammenfassung**

Funktionen & Vorteile von SB-Kassen

- **Funktionen:**
  - Kunden können Artikel selbst scannen und stornieren
  - Anzahl der Artikel kann ausgewählt werden
  - Obst / Gemüse / Backware auswählen und ggf. wiegen
  - Bezahlung mit Karte oder Bargeld
- **Vorteile:**
  - Einsparung von Personalkosten
  - Arbeitserleichterung der Angestellten
  - Steigerung der Kundenzufriedenheit (Vermeidung von Warteschlangen)

Fehlerursachen & Problemstellung

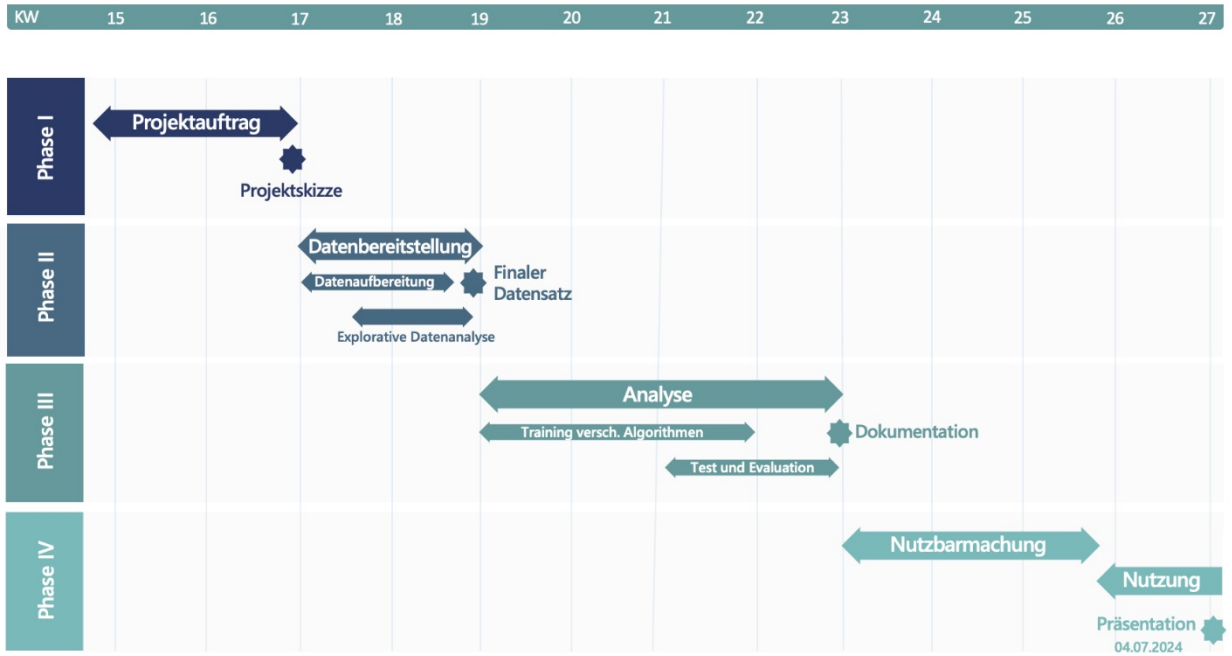
- **Fehlerursachen:**
  - Technische Probleme: Artikel nicht gefunden; Barcode nicht lesbar
  - Versehen: Artikel übersehen; falsche Anzahl oder falsches Produkt gewählt
  - Absicht: Artikel bewusst nicht gescannt; falsche Anzahl oder falsches Produkt gewählt; nachträgliche Stornierung; Abbruch des Zahlungsvorgangs
- **Problemstellung:**
  - Einführung von SB-Kassen im Jahr 2016
  - Anzahl fehlerhafter Bezahlvorgänge seit Inbetriebnahme angestiegen
  - Fehlerhafter Bezahlvorgang: Nicht alle Artikel eines Einkaufs gescannt
  - Empirische Untersuchungen zeigten: ca. 5% der Einkäufe sind fehlerhaft

Projektziel & Anforderungen

- **Ziel:**
  - Fehlerhafte Einkäufe an SB-Kassen erkennen
  - Gewinnsteigerung
  - Zielgerichtete Nachkontrollen
  - Dadurch Verringerung fehlerhafter Einkäufe
- **Anforderungen:**
  - Möglichst viele fehlerhafte Einkäufe erkennen
  - Möglichst wenig Falschverdächtigungen „unschuldiger“ Kunden
  - Maximierung der Gewinnfunktion: **5€ ∗ TP – 25€ ∗ FP – 5€ ∗ FN**

Projektauftrag & Ausgestaltung

- **Auftrag:**
  - Entwicklung eines Klassifikationsalgorithmus zur Erkennung verdächtiger Scanvorgänge (Hinweis an Mitarbeiter zur Nachkontrolle)
  - Entscheidung für Mitarbeiter möglichst nachvollziehbar
- **Ausgestaltung:**
  - Strukturiertes Vorgehen nach DASC-PM
  - Beginn: 08.04.2024
  - Fertigstellung: 04.07.2024



Agenda

- Projektauftrag
- **Datenbereitstellung**
  - Datenvorverarbeitung
  - explorative Datenanalyse
- Analyse
  - Vorgehensweise
  - Details zum endgültigen Modell
  - Mehrwert des Klassifikators
- Nutzbarmachung
- Zusammenfassung

Ursprungsdatenquelle

- 104.646 Einträge (100.105 Einträge *normal*, 4541 *fraud*)
- 12 Attribute, darunter auch das Label (Klassifikation)

	GUID	grand_total	n_items	total_checkout_time	line_voids	most_freq_product	products	timestamp	payment_medium	label	customer_feedback	cash_desk_id
0	8cc21dee-9922-4a41-a9dc-467810ca3db5	63.31	24	324.426177	1	dry	['alcohol', 'fruit and vegetables', 'snack', '...	1483345009517544427	card	normal	NaN	2
1	d80034dc-0087-4546-b5bc-44bd7060c0d5	29.59	11	92.845052	0	dry	['dry', 'fruit and vegetables', 'dry', 'snack'...	1483345752494372926	card	normal	NaN	1

Produktkategorien:
dry, alcohol, fruit & vegetables, snack, energy_drink, bakery, household und convenience

Datenqualität

- **Repräsentativität** für die Filialen und die Umgebung gegeben, nicht jedoch für Deutschland
- **Aktualität** für Testdaten aus 2019 gegeben (Training auf aktuellen Daten für den realen Einsatz empfohlen)
- **Fehlerfreiheit** ist gegeben (keine Duplikate; Ausreißer plausibel)
- **Vollständigkeit** ist für alle Attribute bis auf *customer_feedback* gegeben (letzteres nur 8.348 Einträge)
- **Konsistenz** teilweise nicht gegeben**:**
  - *grand_total* teilweise mit 2 bzw. 3 Nachkommastellen gespeichert
  - *payment_medium* enthält 3 Ausprägungen, obwohl nur zwei Bezahlarten möglich ist
  - *customer_feedback* sollte ganzzahlig zwischen 1 und 10 liegen, aber es gibt Nachkommastellen

Datenaufbereitung

- Entfernung von *GUID*, da irrelevant
- Entfernung von *customer_feedback*, da nur in 8% enthalten und keine sinnvolle Ersetzungsstrategie
- Merkmalserzeugung aus *timestamp: weekday* und *hour* (Rest verworfen)
- Merkmalserzeugung aus *products:* Neue Spalte für jede Produktkategorie mit Angabe der Anzahl
- *total_checkout_time:* Rundung als Ganzzahl, da Milisekunden zu detailliert sind
- *grand_total:* Einträge mit 3 Nachkommastellen auf 2 Nachkommastellen gekürzt
- *payment_medium:* Kartenzahlungen einheitlich als *card* bezeichnet
- Datentransformationen:
  - One-Hot-Codierung von *payment_medium* und *most_freq_product*
  - Binärcodierung von *label*
  - Codierung von *weekday* mit 0 bis 5 für Ausprägungen Montag bis Samstag

Ziele der explorativen Datenanalyse

- Datenvisualisierung
- Statistische Analysen
- Identifikation von Ausreißern

Korrelationen (Pearson)

	label
grand_total	0.144469
n_items	-0.000345
total_checkout_time	0.050646
line_voids	0.002112
payment_medium	0.159123
label	1.000000
cash_desk_id	0.002991

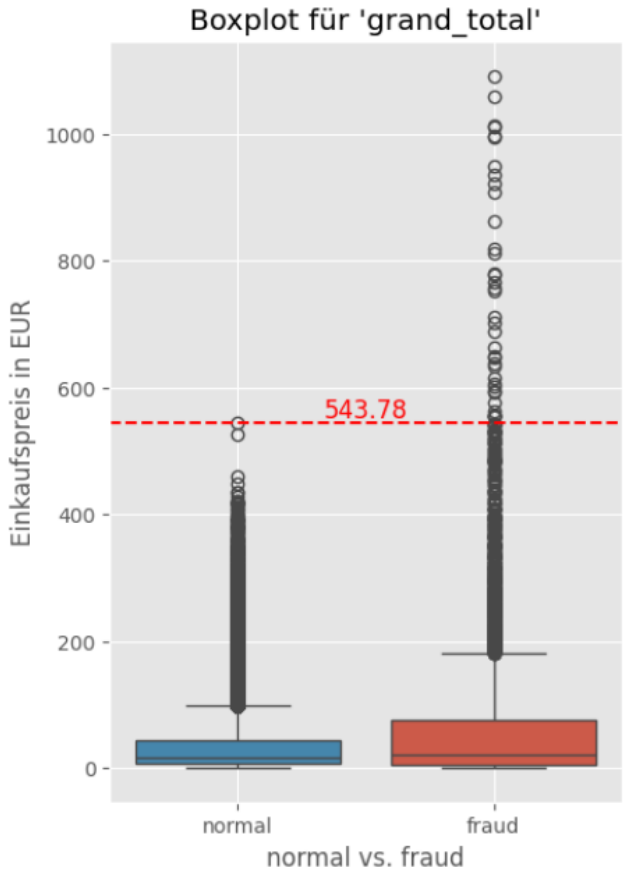
	label
year	0.002246
month	-0.001709
day	0.000231
hour	0.062477
minute	-0.002158
weekday	0.000352
alcohol	0.014530

	label
bakery	-0.071877
convenience	0.095816
dry	-0.035326
energy_drink	0.184902
fruit and vegetables	-0.036250
household	0.236313
snack	-0.071853

- Nur wenige Attribute zeigen eine Korrelation mit label
- Höchste Korrelation: energy_drink und household
- Korrelationen spiegeln sich auch im Modell wider

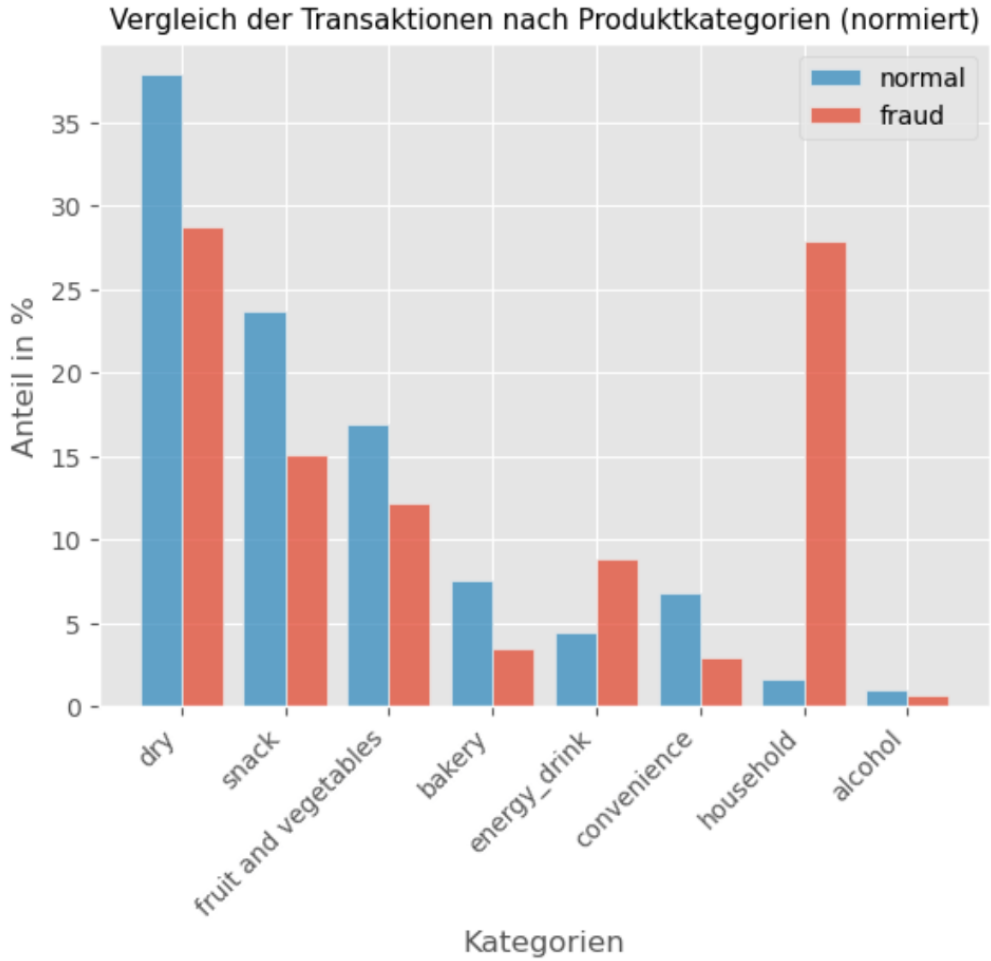
Einkaufswert – *grand_total*

- Die Hälfte der Einkäufe liegt unter einem Warenwert von 20 EUR
- 50%-Perzentil = 17.06 (Median)
- Ausreißer: 60 Einkäufe > 500 EUR
- Maximum 1091 EUR
- Korrelation zum Attribut *label*
- Ausreißer oberhalb von ca. EUR 544 sind *fraud*



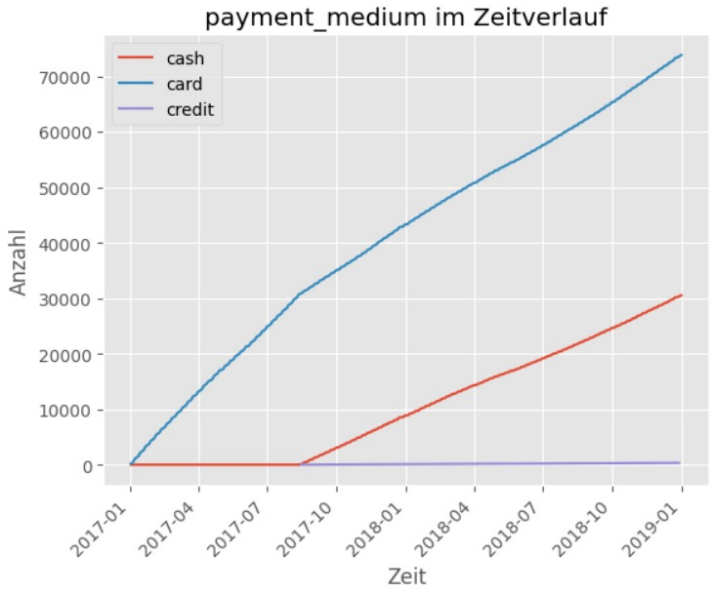
Meistgekauftes Produkt – *most_frequent_product*

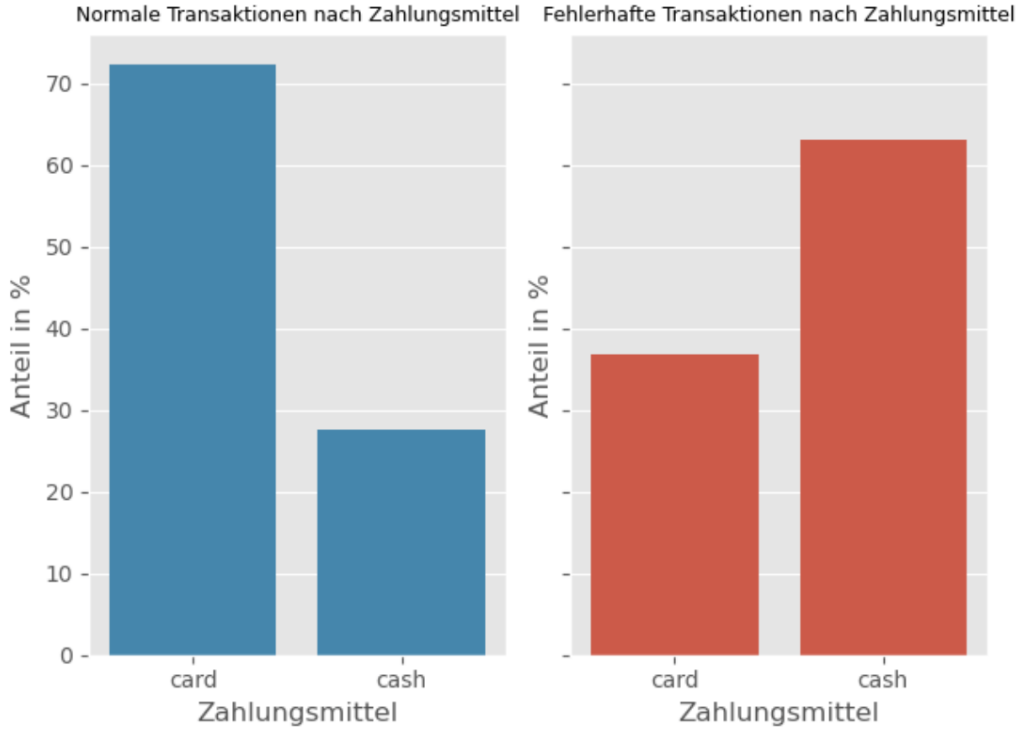
- Kategorische Variable
- Das Attribut *most_frequent_product* weist einen Zusammenhang zu *label* auf
- Bei normalen Transaktionen ist der Anteil von *dry* am größten, bei fehlerhaften Transaktionen *household*
- Relative Häufigkeit von *household* und *energy drink* bei fehlerhaften Transaktionen höher als bei normalen



Zahlungsmittel – *payment_medium*

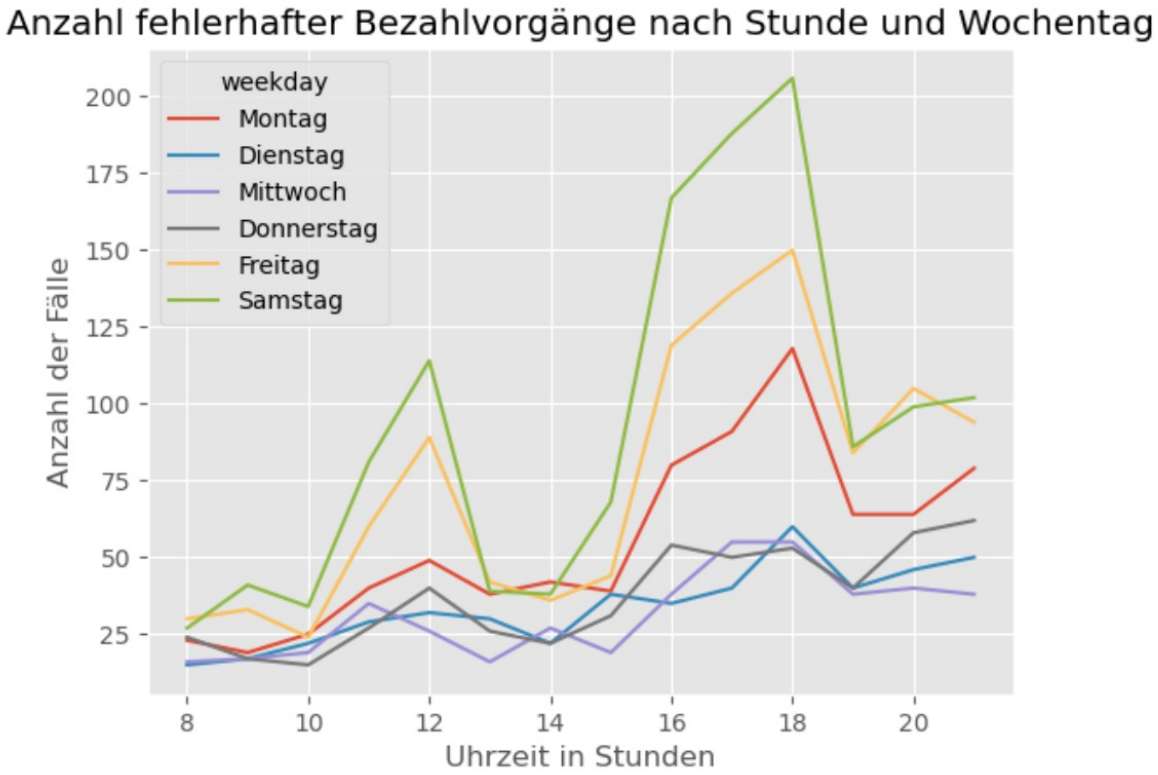
- Kategorische Variable, Ausprägungen: *card, cash* und *credit*
- Bis zum 12.08.2017 nur *card*
- Danach zusätzlich *cash* und *credit*





Zeitlicher Zusammenhang – *timestamp*

- Korrelation zwischen Stunde und *label*
- Korrelation zwischen Wochentag und *label*
- Auffällig:
  - Häufung von *fraud* freitags und samstags
  - Zwischen 16 und 19 Uhr deutlicher Anstieg von *fraud*

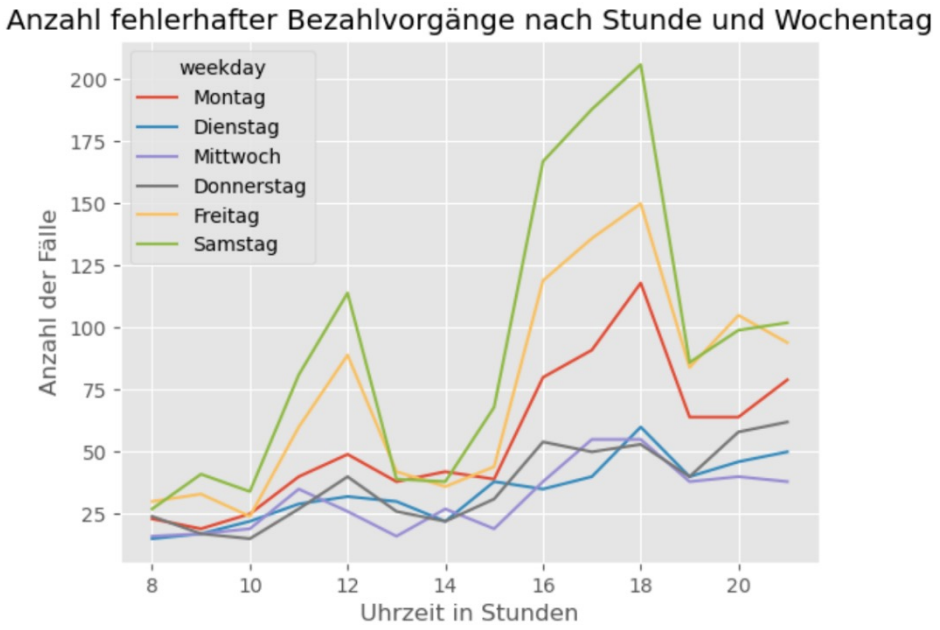


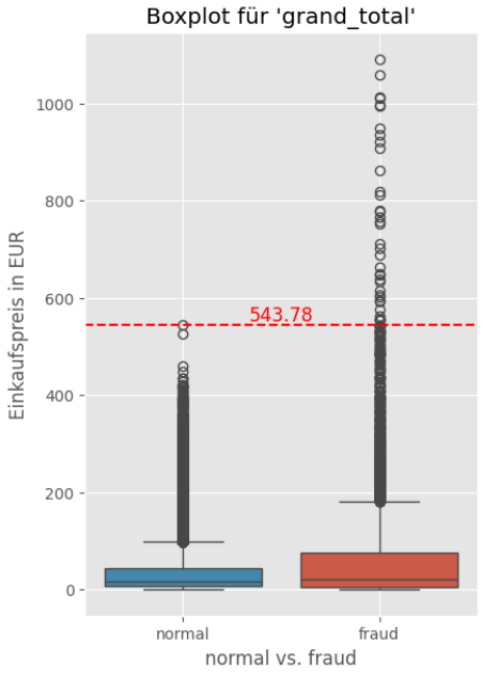
Nicht-technische Umsetzungsmöglichkeiten

- Begrenzung des Gesamtpreises bei 500 EUR

  In dem Fall würden im Datensatz 58 Betrugsfälle wegfallen (aber auch 2 normale Einkäufe)

- Überwachung der SB-Kassen an Freitagen und Samstagen zwischen 16 und 19 Uhr





Agenda

- Projektauftrag
- Datenbereitstellung
  - Datenvorverarbeitung
  - explorative Datenanalyse
- **Analyse**
  - Vorgehensweise
  - Details zum endgültigen Modell
  - Mehrwert des Klassifikators
- Nutzbarmachung
- Zusammenfassung

Vorgehensweise

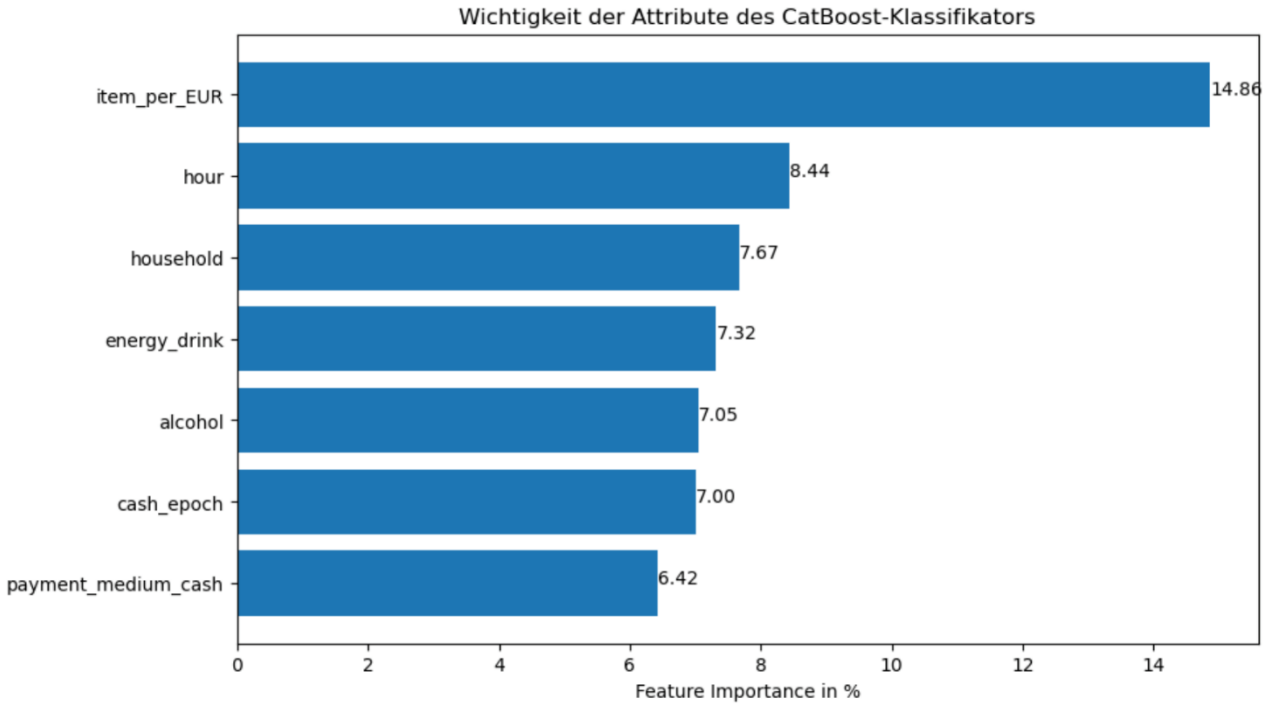
- Alle Modelle aus dem „scikit-learn-Universum“ wurden getestet
- Vorläufiger Sieger: Random Forest (RF)
- Fokus auf Algorithmen, die auf Entscheidungsbäumen basieren
- Weitere Tests mit Boosting Algorithmen
- Boosting Algorithmen lieferten noch bessere Ergebnisse als RF
- CatBoost war unter den Boosting Algorithmen der beste Algorithmus

Vorgehensweise

- Feature Selection anhand von CatBoost:
  - Zusätzliche Attribute wie Kundendichte, Feiertage usw. getestet und verworfen
  - Zusätzliche Attribute, die wertvolle Informationen enthalten (z.B. *cash_epoch* und *item_per_time*), identifiziert
  - Vielzahl von Ratios von Attributen ausprobiert und verworfen
  - Korrektur unausgeglichener Daten ausprobiert und verworfen
- Hyperparameter-Tuning mit CatBoost
- Trainieren des besten Modells auf kompletten Datensatz
- Vorhersage auf Testdaten aus 2019

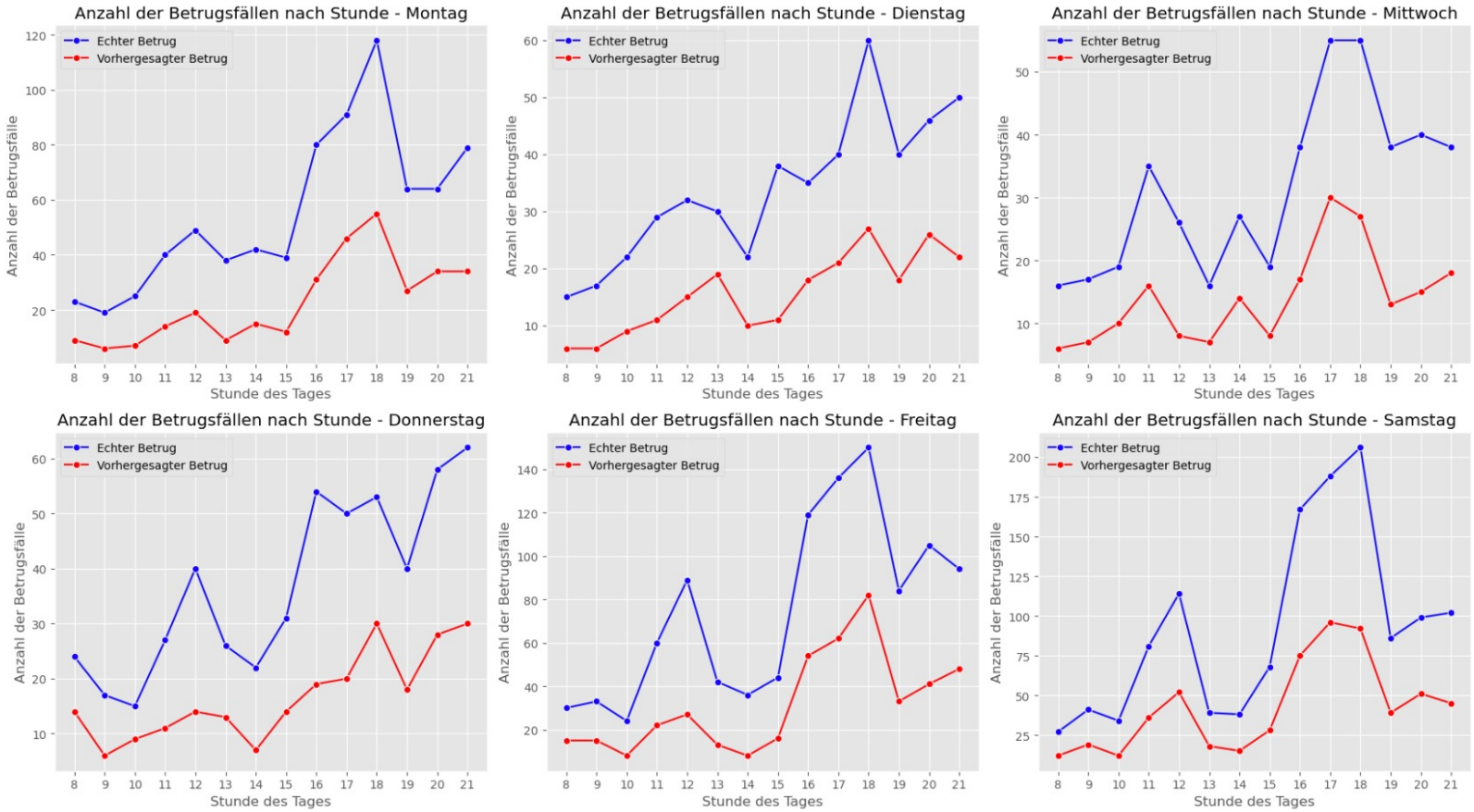
Details zum endgültigen Modell

- Möglichkeit, sogenannte *Feature Importances* anzeigen zu lassen
- Werte sind normiert und können als Prozent-Za interpretiert werden
- Addition aller Feature Importances ergibt 100%
- **Niedriger Wert:** Änderung der Attributausprägu (Feature Value) beeinflusst die vorhergesagte Wahrscheinlichkeit für die positive Klasse im Durchschnitt weniger stark.
- **Hoher Wert:** Änderung der Attributausprägung beeinflusst die vorhergesagte Wahrscheinlichkeit für die positive Klasse im Durchschnitt stark.



Wichtigkeiten der Attribute nach Optimierung

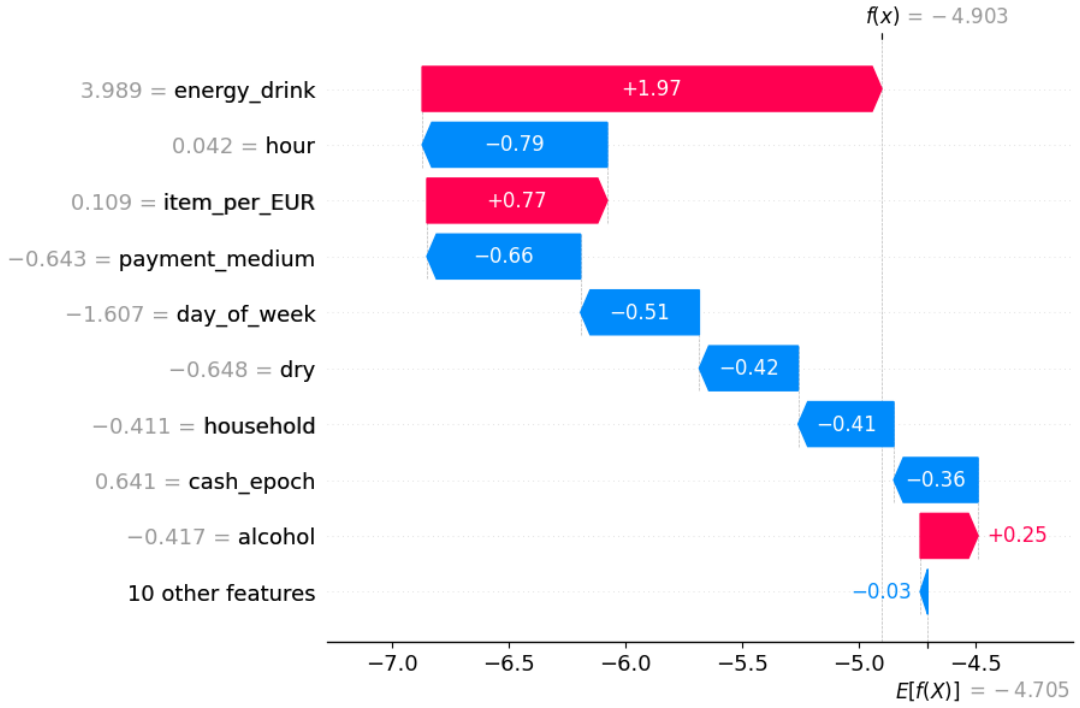
Das Modell lernt das Zusammenspiel mehrerer Attribute

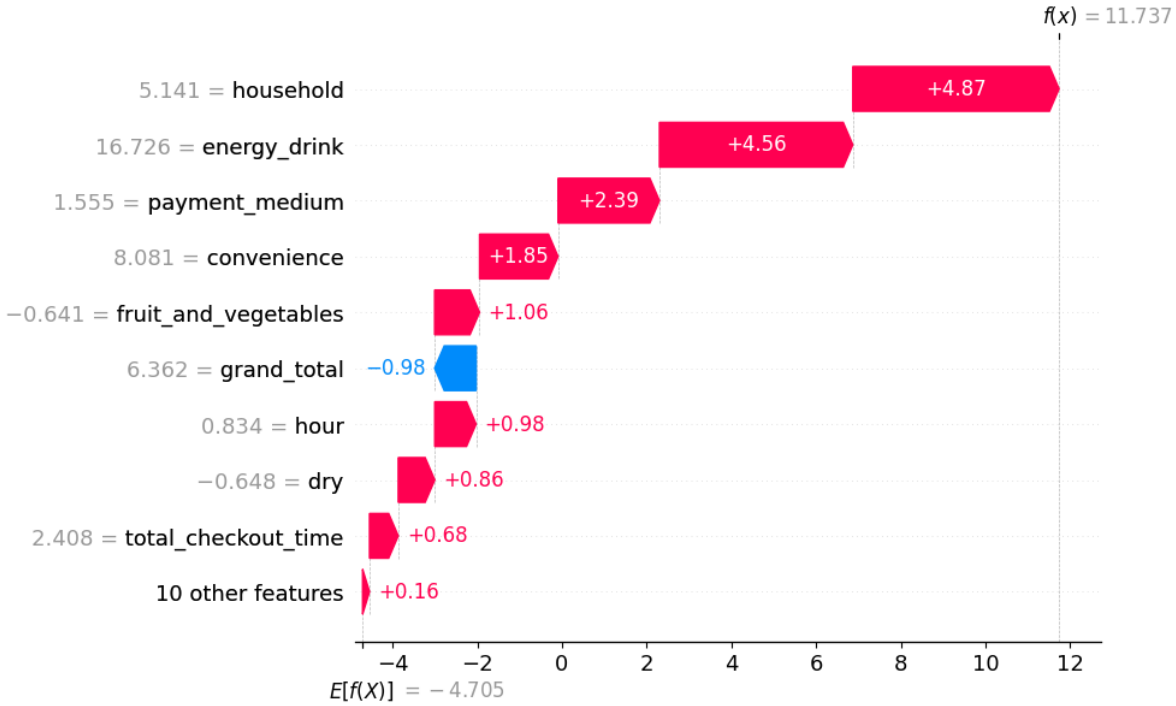


SHAP Values

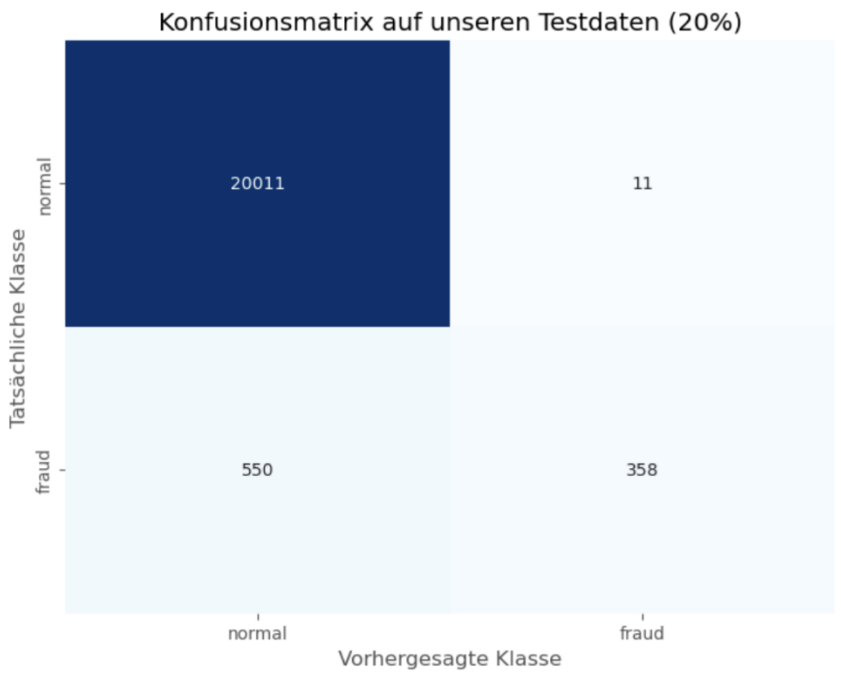
- Nach Lloyd Shapley (1953) aus der kooperativen Spieltheorie
- Misst den durchschnittlichen zusätzlichen Beitrag einer Variablen
- Beispiel: Wohnung wird auf 310k EUR geschätzt anhand von 3 Attributen:
  - 50qm
  - "Nähe Park“
  - 2. Stock
- Was ist der Beitrag von „Nähe Park“ zum Kaufpreis?
- Schätze den Wert der Wohnung mit der Kombination aller Attribute einmal mit und einmal ohne „Nähe Park“, bestimme jeweils die Differenz und bilde den Durchschnitt der Differenzen

Erklärung einer Klassifikation mittels SHAP Values

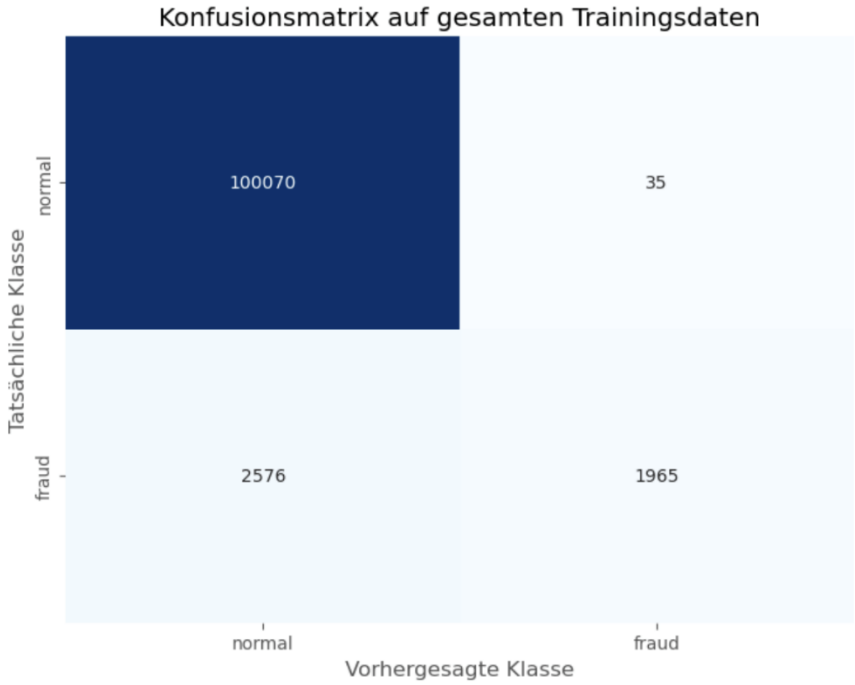




Ergebnisse auf Trainings- und Testdaten



Gewinn ohne Modell: - 4540 €
Gewinn mit Modell: - 1235 €
Mehrwert: 3305 €

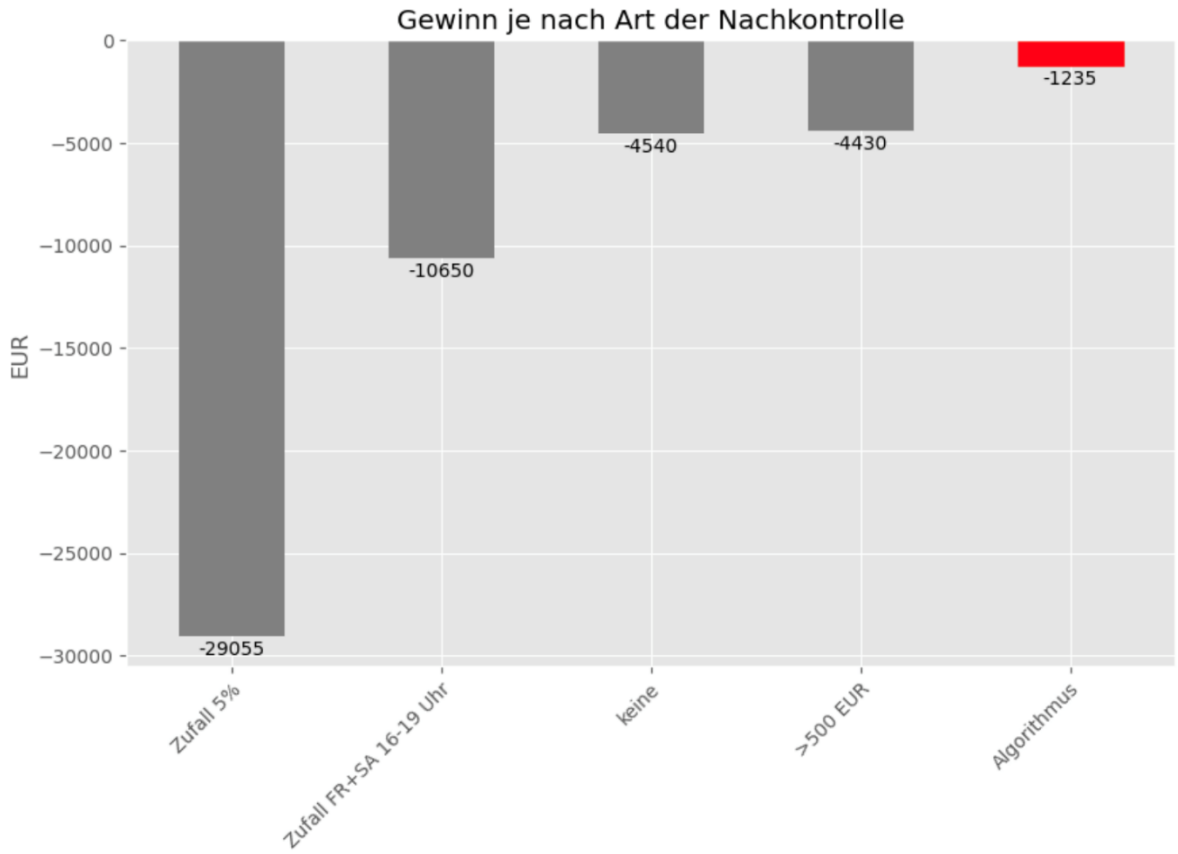


Gewinn ohne Modell: - 22705 €
Gewinn mit Modell: - 3930 €
Mehrwert: 18775 €

Mehrwert

Vergleich zwischen:

- Zufallskontrollen (5% der Einkäufe)
- Zufallskontrollen an Freitagen und Samstagen zwischen 16 und 19 Uhr (davon 10% der Einkäufe)
- Keine Kontrollen
- Kontrolle aller Einkäufe > 500 Euro
- **Unser Vorhersagemodell**

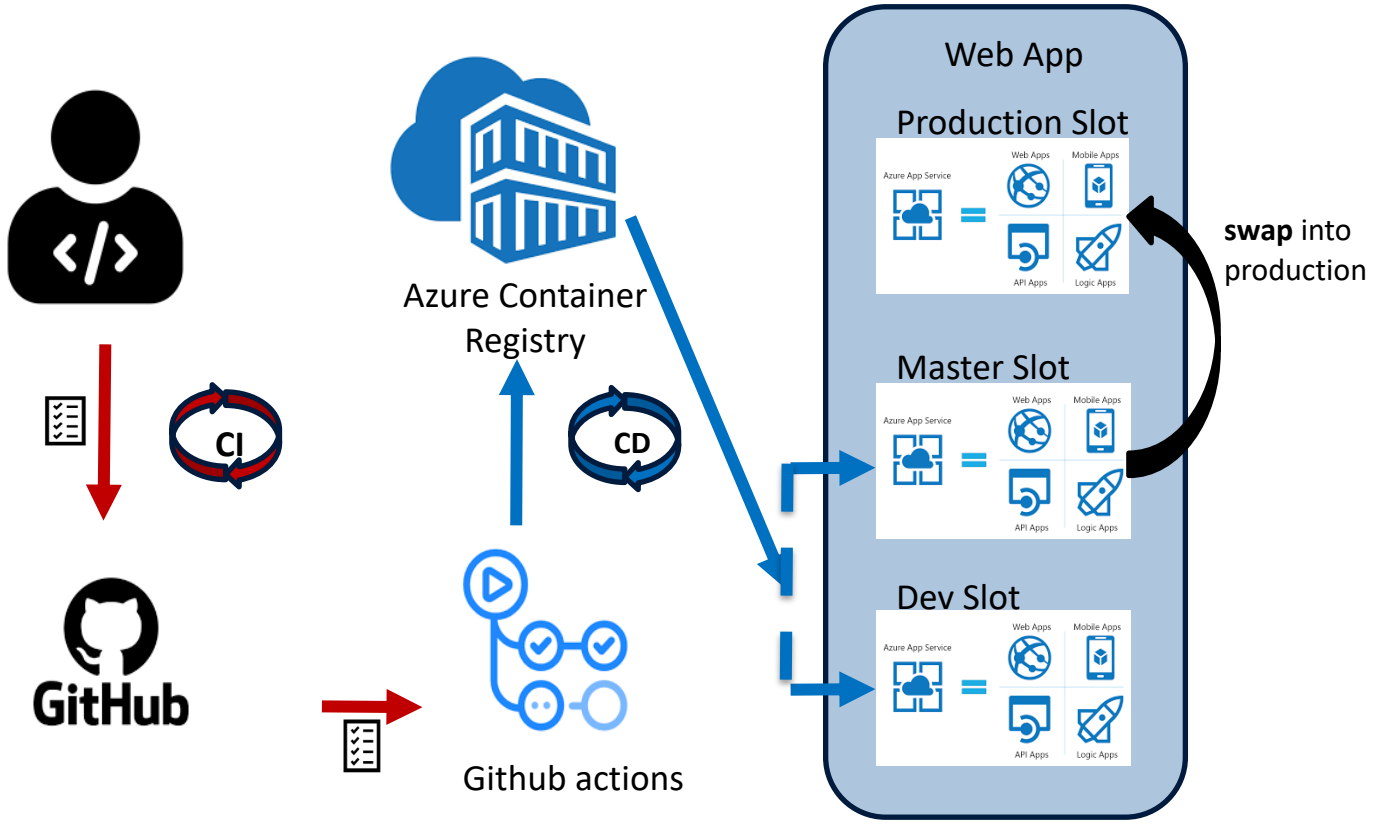


Verwendung der Gewinnfunktion
5€ * TP – 25€ * FP – 5€ * FN

Agenda

- Projektauftrag
- Datenbereitstellung
  - Datenvorverarbeitung
  - explorative Datenanalyse
- Analyse
  - Vorgehensweise
  - Details zum endgültigen Modell
  - Mehrwert des Klassifikators
- **Nutzbarmachung**
- Zusammenfassung

Nutzbarmachung



Nutzbarmachung

- Code wird auf GitHub entwickelt, Rest-API hosted auf Azure
- Qualitätskontrolle:
  - Trennung von Produktion und Entwicklung
  - Tests auf Master-Branch: Nur bei Erfolg ist Code-push möglich
- Ausfallsicherheit:
  - Web-App hat mehrere Instanzen mit automatischem load-balancing
  - Hot-start der Produktionsinstanz (keine downtime)
  - Kontinuierliche health-checks der laufenden Instanzen
- Skalierbarkeit der Anwendung ist gegeben
- Email Benachrichtigung bei unbekannten Produkten und Zahlungsmethoden, um schnell auf Änderungen reagieren zu können (optional)

Agenda

- Projektauftrag
- Datenbereitstellung
  - Datenvorverarbeitung
  - explorative Datenanalyse
- Analyse
  - Vorgehensweise
  - Details zum endgültigen Modell
  - Mehrwert des Klassifikators
- Nutzbarmachung
- **Zusammenfassung**

Zusammenfassung

- Auftrag: Erstellung eines Klassifikationsalgorithmus, der mit möglichst hoher Präzision anzeigt, welche Einkäufe an den SB-Kassen nachkontrolliert werden sollen
- Transaktionsdaten enthalten genug Informationen, um einen solchen Algorithmus zu erstellen
- Ein Boosting-Algorithmus konnte auf Trainingsdaten nachweisen, dass sein Einsatz ökonomisch sinnvoll ist und verschiedene Heuristiken wie Zufallskontrollen oder Kontrollen zu bestimmten Uhrzeiten schlägt
- Erklärungen für die jeweilige Entscheidung des Algorithmus (SHAP Values) werden zusätzlich geliefert, um Transparenz bei Nachkontrollen zu schaffen
- Implementierung mittels REST-API erlaubt eine relativ effiziente und wartungsarme Umsetzung, der Algorithmus kann zukünftig selbstständig mit neuen Daten trainiert werden

Vielen Dank für die Aufmerksamkeit!