



# Optimierung der Newsletter Öffnungsrate Reinforcement Learning

Simon Schärer

December 6, 2024



# Agenda

1. Ausgangslage
2. Idee
3. Datengrundlage
4. Verteilung Trainingsdatensatz
5. Modellübersicht
6. Modell-Details
7. Leistung
8. Klassifizierung der Testdaten
9. Learnings / Code



# Ausgangslage

Ich versende monatlich einen Newsletter an ca. 30'000 Kunden mit dem Ziel, dass die Kunden den Newsletter lesen und mit uns interagieren. Als Interaktion zählt nicht nur der Kauf eines Produktes, sondern für mich ist auch schon der Klick auf Text, Blogeintrag etc. wertvoll, damit ich ein Kundenprofil erstellen kann und somit mehr Informationen (Interessen) über den Kunden erfahre.

Der Erfolg eines Newsletter kann mittels Öffnungsrate, Klickrate etc. gemessen werden.

Aus meiner vergangen Tätigkeit weiss ich, dass nur schon der Betreff Einfluss auf die Öffnungsrate hat.



In meinem Beispiel hat jeder Kunden eines der folgenden Interessen:

- ▶ Anlegen
- ▶ Finanzieren
- ▶ Vorsorge

Wenn ich den Betreff des Newsletters richtig, also dem Kunden entsprechend richtig treffe, öffnet er den Newsletter und ist für mich ein Erfolg.



# Datengrundlage

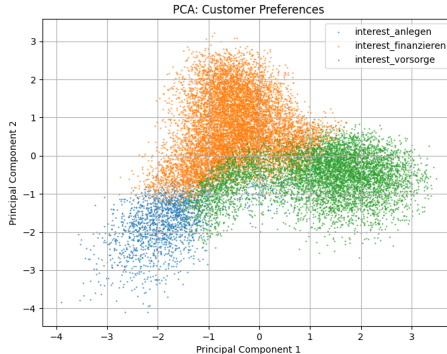
Da ich für dieses Experiment keine echten Daten nutzen kann, habe ich alle Daten selber generiert. Dies stellte sich als grössere Herausforderung heraus als ursprünglich gedacht.

Alter	Finanzieren	Anlegen	Vorsorge	Groundtruth
34	0.432	0.343	0.364	Finanzieren
56	0.128	0.531	0.358	Anlegen
45	0.985	0.365	0.663	Vorsorge

Table: Datengrundlage



# Verteilung Trainingsdatensatz



## Klarer Unterschied

Ich musste sicherstellen, dass der RL Algorithmus auch etwas lernen kann, sonst wusste ich nicht, ob die Daten oder mein Algorithmus schlecht ist. Deshalb habe ich meine Daten so erstellt, dass sie klar unterschiedlich sind.



- ▶ **Ansatz:** Modell mit einem DQN-basierten Agenten (Deep Q-Network).
- ▶ **Umgebung:** Simulation von Betreff mit drei Finanzthemen:
  - ▶ **Anlegen, Finanzieren, Vorsorge.**
- ▶ **Zustandsrepräsentation:**
  - ▶ Merkmale: Alter, finanzielle Interessen (Anlegen, Finanzieren, Vorsorge)
  - ▶ Standardisiert (standardscaler) für bessere Repräsentation.
- ▶ **Aktionen:** Auswahl eines der drei Finanzthemen basierend auf vorhergesagten Präferenzen.



## Agent:

- ▶ **Architektur:** Deep Q-Network (Policy- und Zielnetzwerke).
- ▶ **Hyperparameter:**
  - ▶ Lernrate: 0.001
  - ▶ Diskontfaktor ( $\gamma$ ): 0.95
  - ▶ Epsilon-Verfall: 0.9
  - ▶ Ziel-Netzwerk-Update-Frequenz: alle 10 Schritte
- ▶ **Reward Funktion:**
  - ▶ Richtiges Thema: +3
  - ▶ Falsches Thema: -1



# Leistung

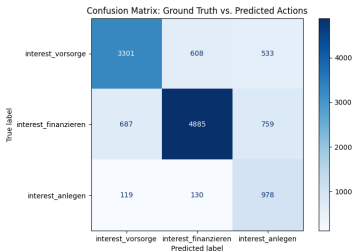


Figure: Absolute Werte

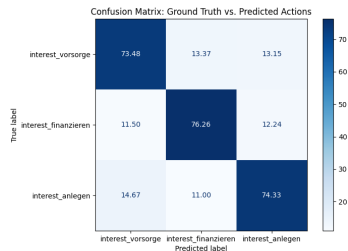


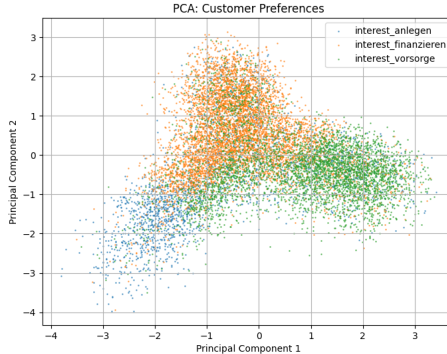
Figure: Relative Werte

## Leistung

- ▶ **Metriken:** Konfusionsmatrix, Accuracy (ca. 75% mit optimierten Parametern).
- ▶ **Bewertung:** Vergleich der vorhergesagten Aktionen mit Ground Truth.



# Klassifizierung der Testdaten



## Testdaten

Die Cluster und die Interessen Zugehörigkeit sind ersichtlich und stimmen grösstenteils.



## Learnings:

- ▶ Ziel muss klar und am Anfang einfach sein, musste die Zielkomplexität mehrfach nach unten anpassen.
- ▶ Daten waren aufwändig zu generieren.
- ▶ Architektur, Hyperparameter, Reward-Funktion... es gibt viele Stolpersteine.
- ▶ ohne ChatGPT wäre ich immer noch am debuggen...

## Code:

- ▶ [https://github.com/ssch-fpv/lr\\_newsletter](https://github.com/ssch-fpv/lr_newsletter)