Trash-ML: KI Analyse von Stadtreinigungs Daten zur Optimierung von Einsatzplänen

To Uyen Nguyen Thi | Viet Duc Hoang | Justin Pallas | Krist Baliev

{x} Gliederung

1. Problemstellung
    1.1. Problemfrage
    1.2. Ziel und Methode
2. Datenrecherche
    2.1. Planung der Recherche
    2.2. Datenbeschaffung
3. Korrelation zwischen Daten
4. Aggregation der Spalten
5. Modellentwicklung & Evaluierung
6. Fazit und Verbesserungsmöglichkeiten

Berliner Stadtreinigung (Logo)

{1.1} Problemfrage

- Bedarf einer **Optimierung** der Einsatzpläne des **BSR** [Berliner Stadtreinigung]

- Welche **Daten** / **Features** nehmen Einfluss auf die Höhe der **Tonnage** bei der Müllsammlung?

Bild 1: Mülli nicht abgeholti :(

{1.2} Ziel und Methode

Primäres Ziel

- Bereitstellung eines **Prototyp Modells** zur Vorhersage von anfallendem Müll für einen jeweiligen Tag
- Training basierend auf **internen** und **externen** Daten
- Modelltraining mit bewährten **Machine Learning Algorithmen** (z.B. **Regression**)

Methode

1. **Planung** der Datenrecherche
2. **Finden** & **Aufbereiten** von externen Daten
   [Recherche, Lag-Einbau, Korrelation]
3. **Aggregation** von Spalten
4. Modellentwicklung
   [Normalisierung, Algorithmuswahl]
5. **Evaluierung**
6. Eventuelle **Modifizierungen**

{2.1} Planung der Recherche

Planung der Recherche durch **Vermutungen / Hypothesen**

- Wetterdaten [Temperatur, Niederschlag, Schneefall]
- Einwohnerzahlen [Wachstum, Wanderung, Geburten/Tode]
- Umsatz im Einzelhandel/Gastgewerbe
- Ferien und Feiertage
- Verbraucherpreisindex
- Tourismus [Übernachtungen, Ankünfte]

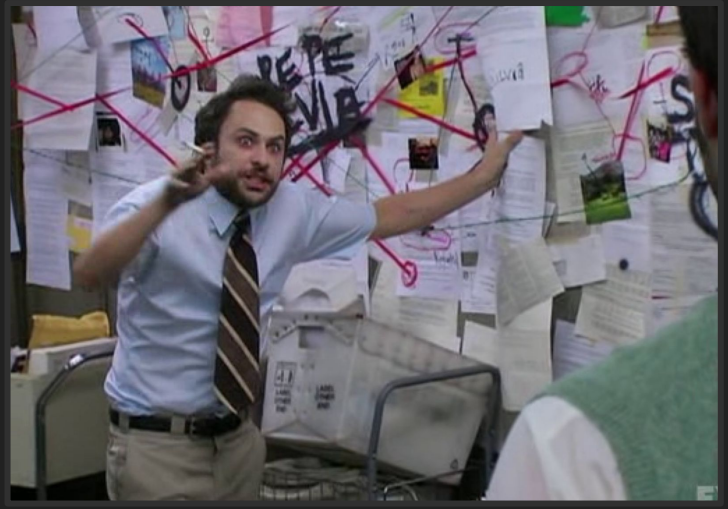


Bild 2: Datenbeschaffung 2025, coloriert

{2.2} Datenbeschaffung

- Statistik Berlin Brandenburg
  [Einzelhandel]
- Meteostat Python Bibliothek (Wetterdaten API)
  [avg_temperature, precipitation, snow(mm)]
- Genesis / Destatis
  [Verbraucherpreisindex, Tourismus]
- Schulferien-online.de
  [Ferien, Feiertage]
- Daten.berlin.de
  [Retail_Offers]
- Statistische Bibliothek
  [Einwohnerzahlen]

```python
from datetime import datetime
from meteostat import Daily
import ssl

# Set SSL context.
ssl._create_default_https_context = ssl._create_unverified_context

# Set time period.
start = datetime(2018, 12, 29)
end = datetime(2023, 12, 30)

# Get daily data.
weather_data = Daily('10381', start, end)
weather_data = weather_data.fetch()
```

Bild 3: Wetterdaten durch Meteostat API

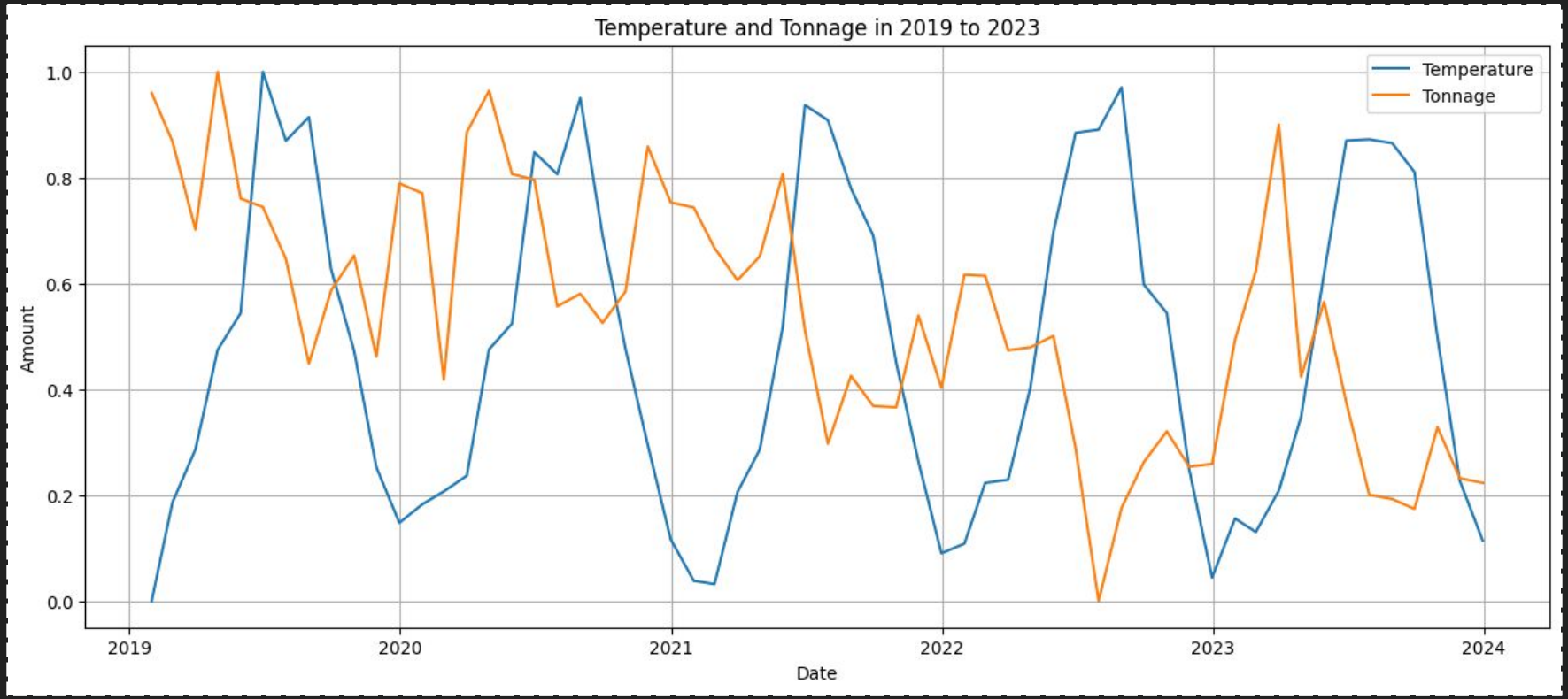
{2.2} Datenaufbereitung

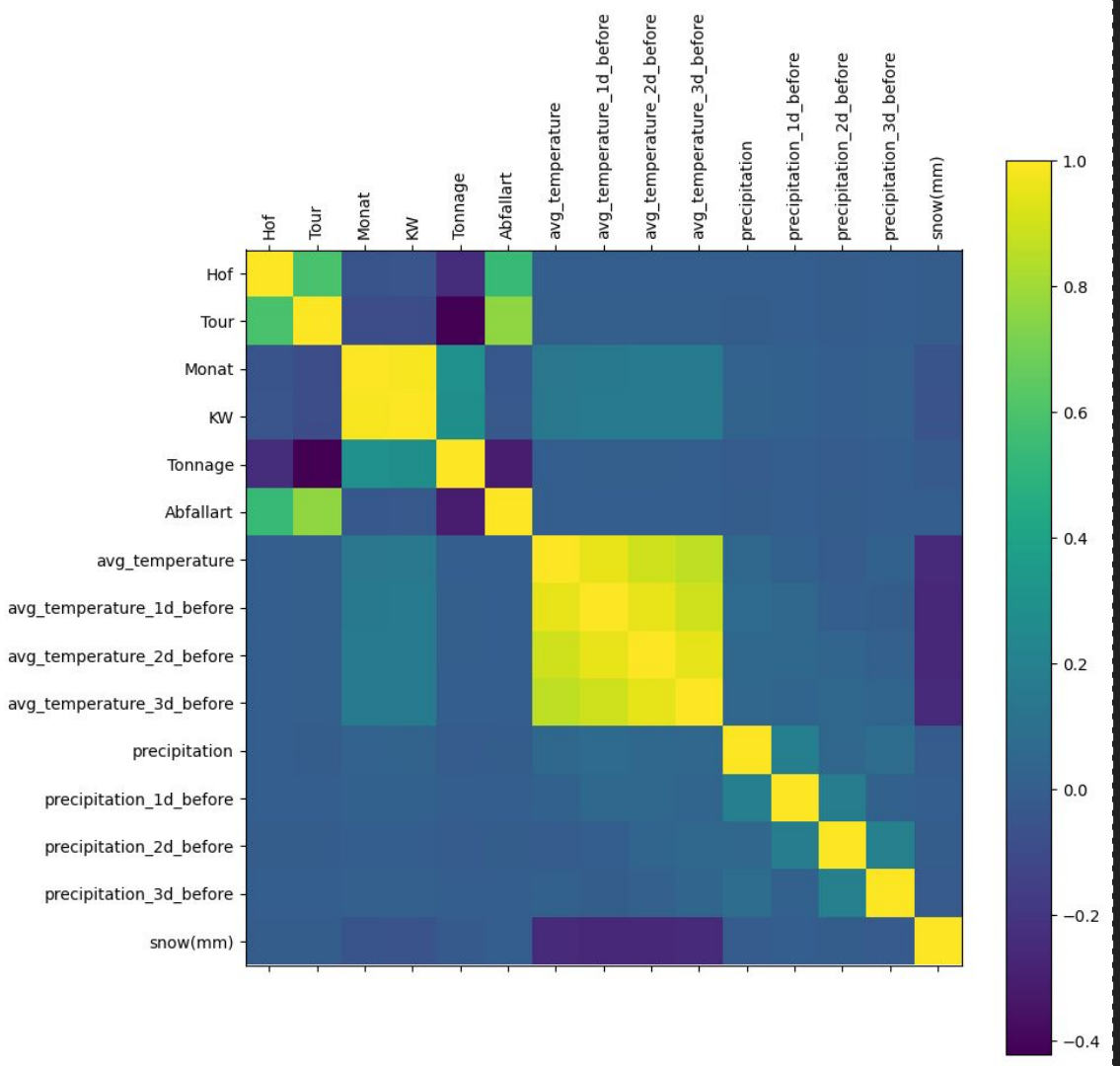
-
- Nan-Values entfernen
- Outlier entfernen
- Datentypen vereinheitlichen
- Mapping auf Spalte Datum

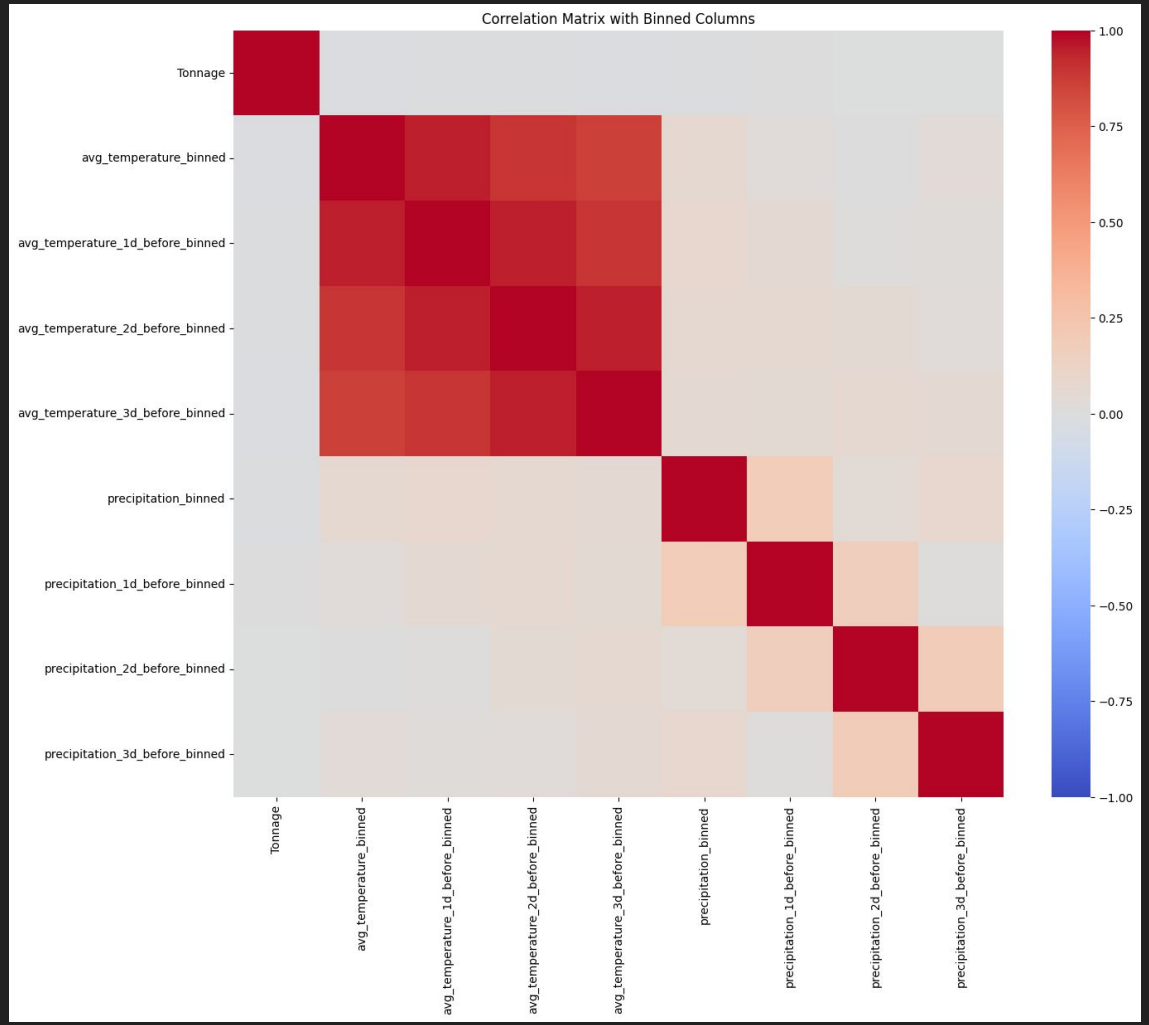
Date	Month	CW	Year	Yard	Shift	Tour	Tonnage
13.02.23	602	2107	608923	513	308	44472	5642.73
13.03.23	948	3476	639268	530	323	46916	6805.29
14.02.23	618	2163	625107	529	316	44543	5553.43
14.03.23	960	3520	647360	547	327	44691	7013.56
27.03.23	948	4108	639268	530	323	42526	6091.06
28.03.23	969	4199	653429	550	330	44689	6010.49

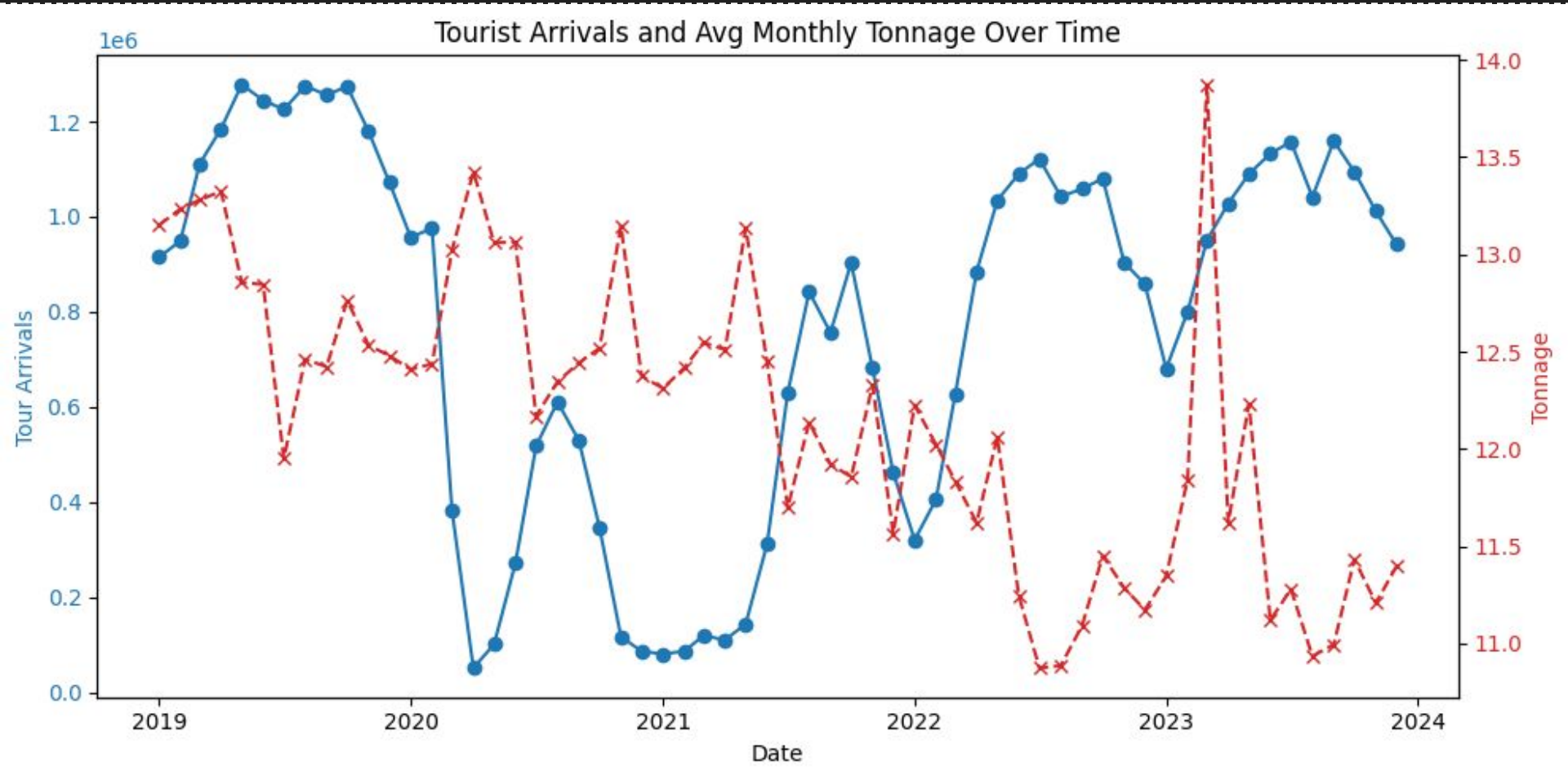
Tage mit einer Tonnage von größer 5500

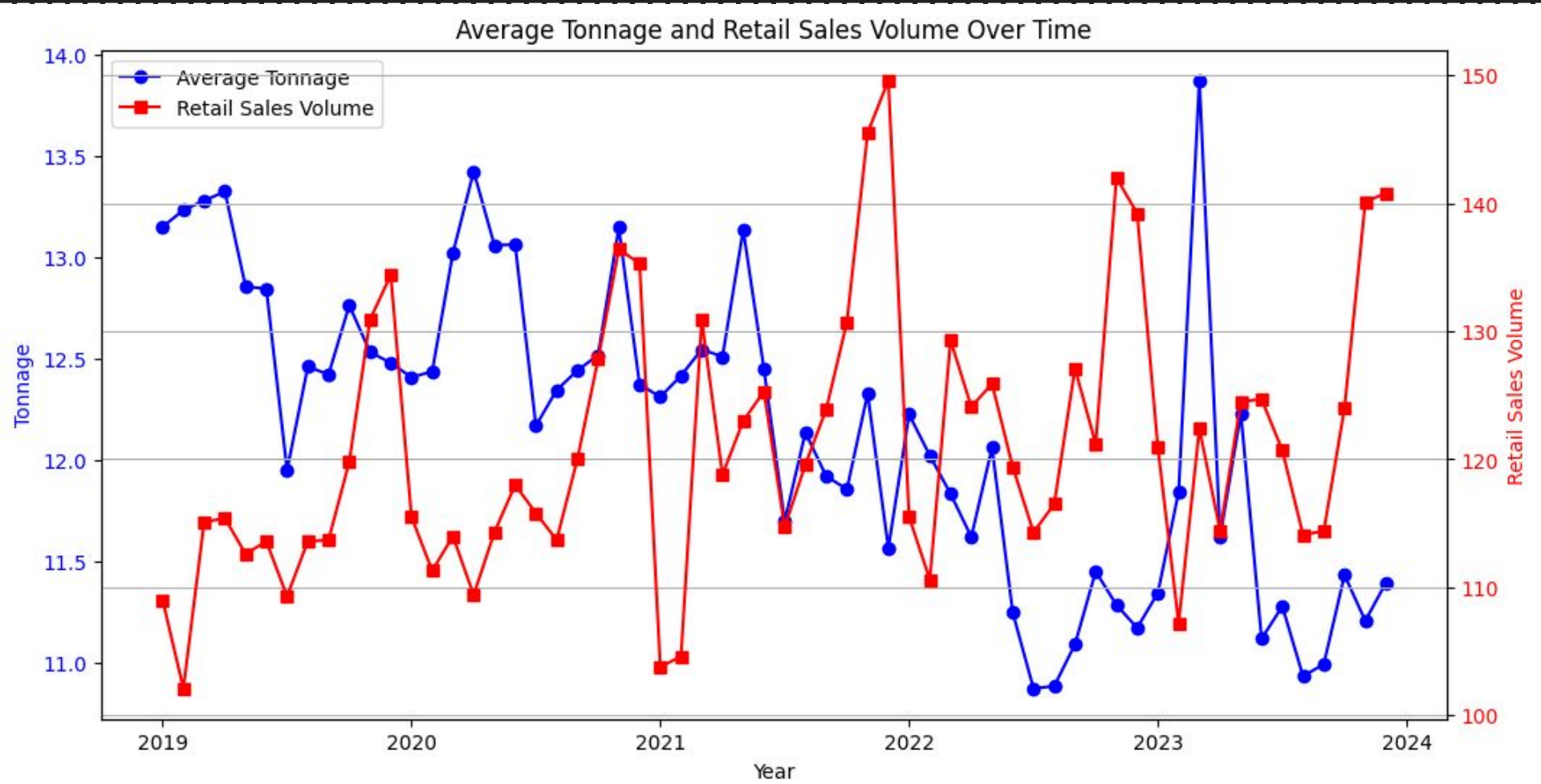
{3} Korrelation zwischen Daten

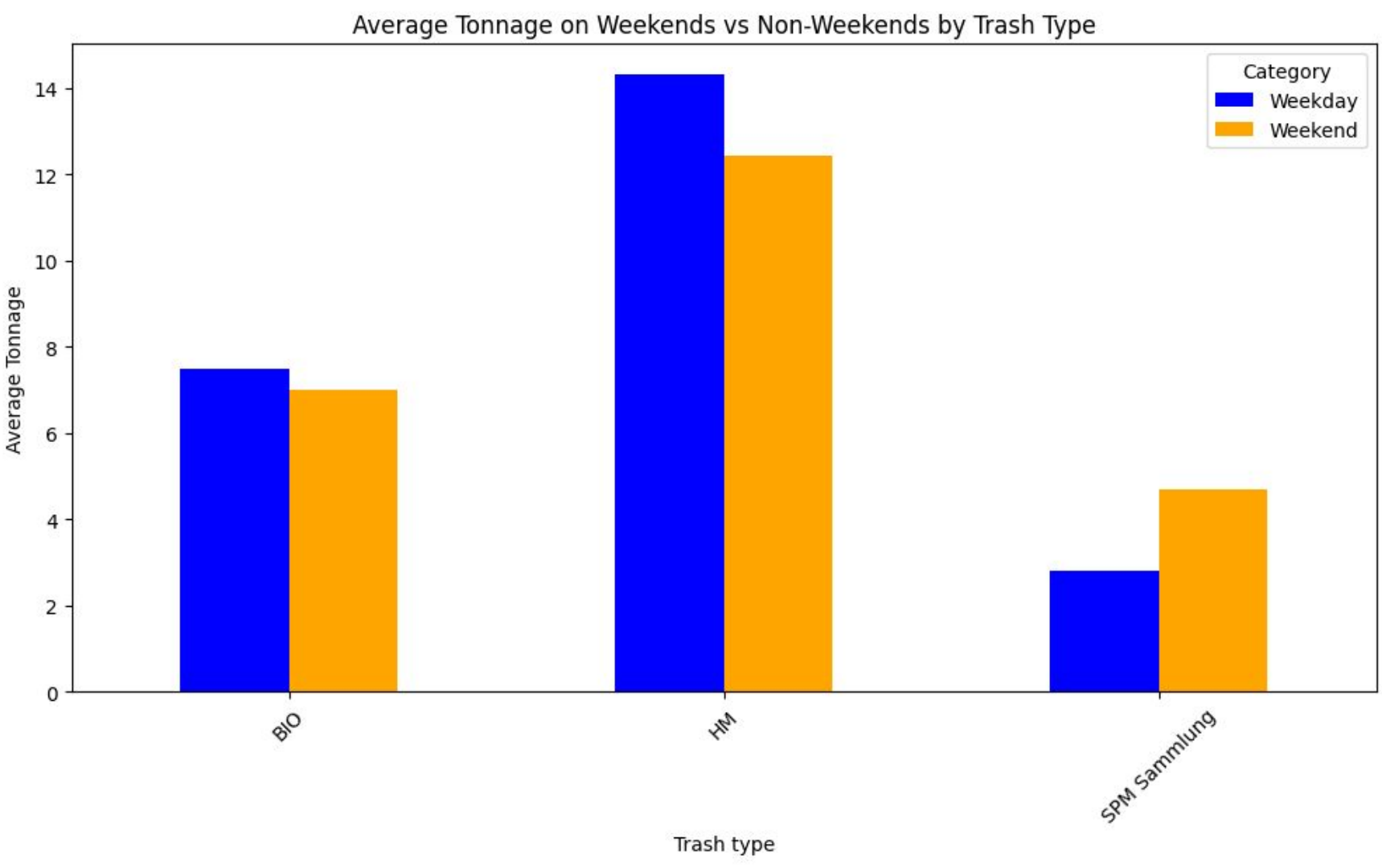


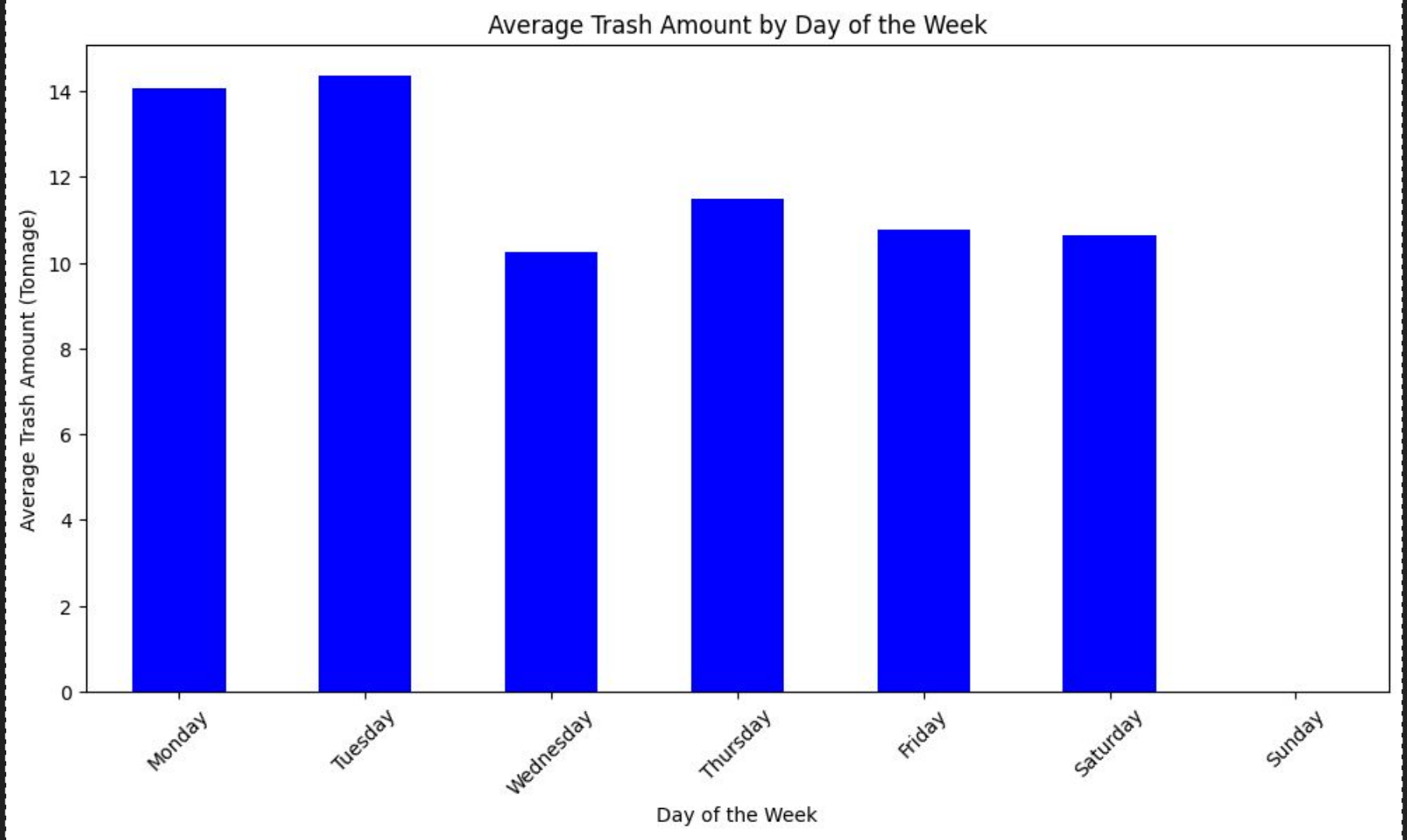

Hof
Tour
Monat
KW
Tonnage
Abfallart
avg_temperature
avg_temperature_1d_before
avg_temperature_2d_before
avg_temperature_3d_before
precipitation
precipitation_1d_before
precipitation_2d_before
precipitation_3d_before
snow(mm)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

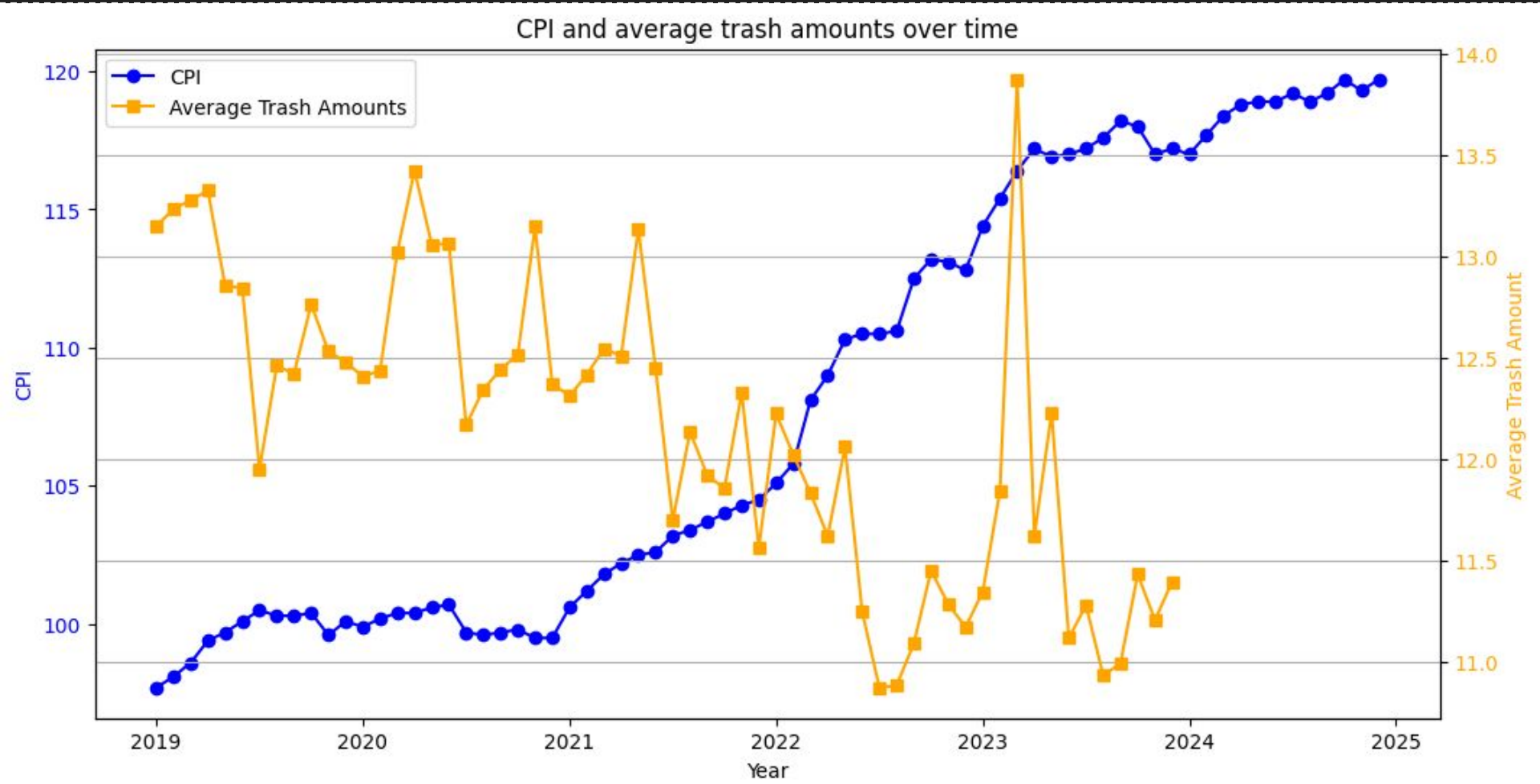

Correlation Matrix with Binned Columns
Tonnage
avg_temperature_binned
avg_temperature_1d_before_binned
avg_temperature_2d_before_binned
avg_temperature_3d_before_binned
precipitation_binned
precipitation_1d_before_binned
precipitation_2d_before_binned
precipitation_3d_before_binned
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

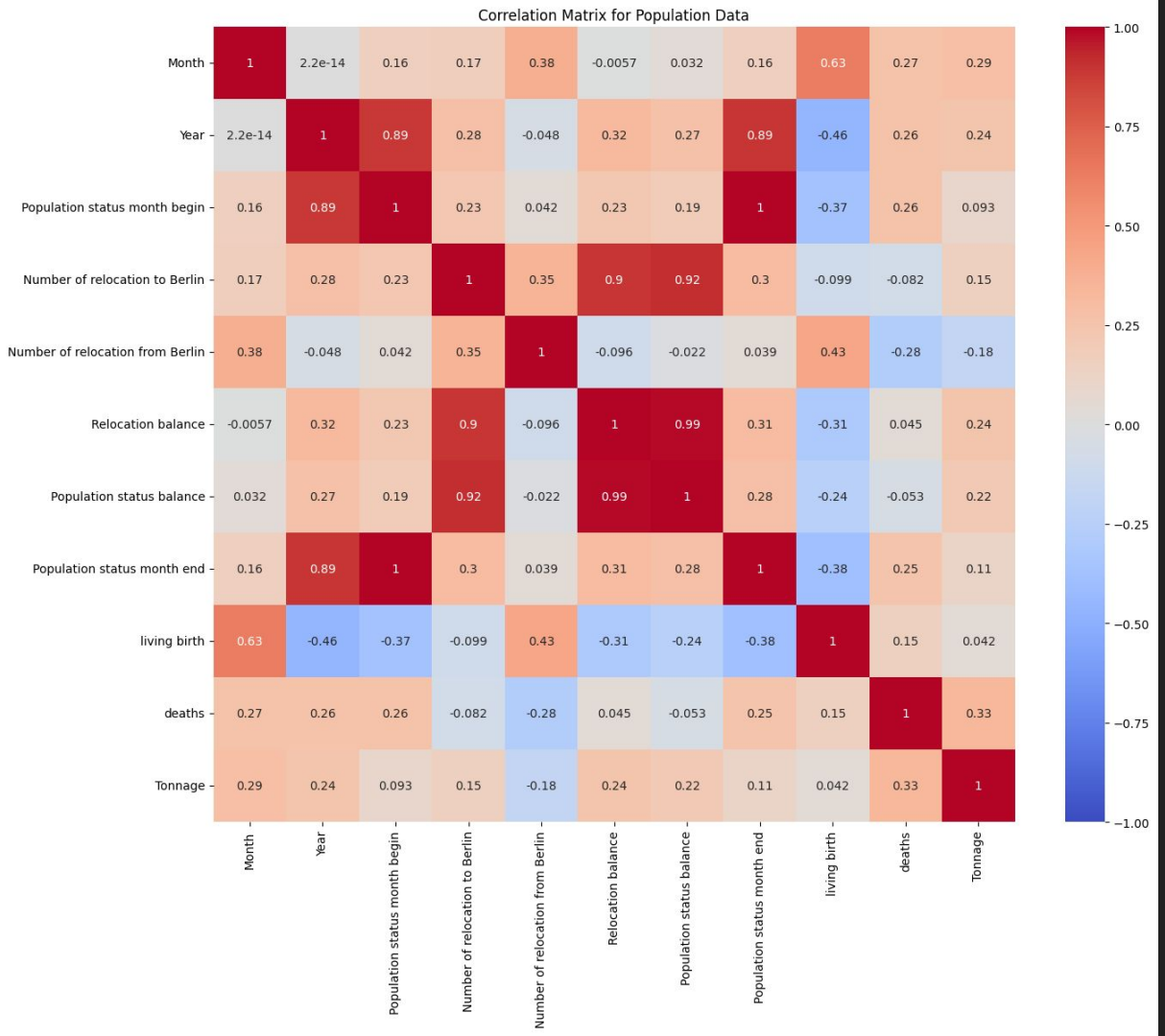

Tourist Arrivals and Avg Monthly Tonnage Over Time
1e6
Tour Arrivals
Tonnage
Date
0.0
0.2
0.4
0.6
0.8
1.0
1.2
11.0
11.5
12.0
12.5
13.0
13.5
14.0
2019
2020
2021
2022
2023
2024


Average Tonnage and Retail Sales Volume Over Time
Average Tonnage
Retail Sales Volume
Tonnage
Retail Sales Volume
Year
14.0
13.5
13.0
12.5
12.0
11.5
11.0
150
140
130
120
110
100
2019
2020
2021
2022
2023
2024

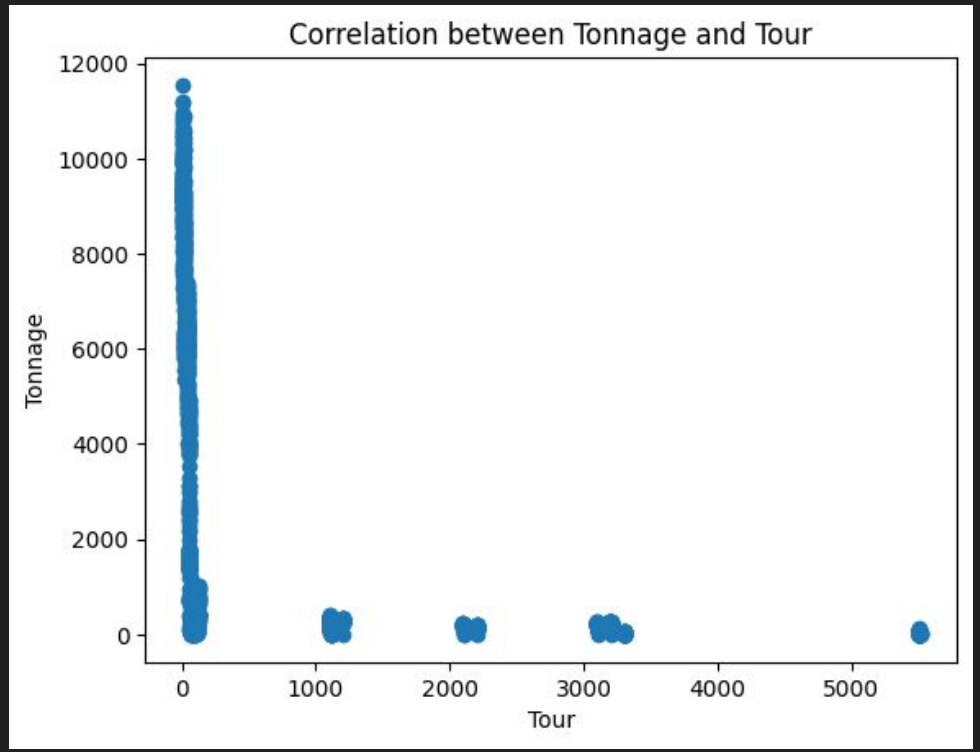

Average Tonnage on Weekends vs Non-Weekends by Trash Type
Category
Weekday
Weekend
Average Tonnage
0
2
4
6
8
10
12
14
BIO
HM
SPM Sammlung
Trash type


Average Trash Amount by Day of the Week
Average Trash Amount (Tonnage)
0
2
4
6
8
10
12
14
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of the Week


CPI and average trash amounts over time
CPI
Average Trash Amounts
120
115
110
105
100
14.0
13.5
13.0
12.5
12.0
11.5
11.0
CPI
Average Trash Amount
2019
2020
2021
2022
2023
2024
2025
Year


Correlation Matrix for Population Data
Month 1 2.2e-14 0.16 0.17 0.38 -0.0057 0.032 0.16 0.63 0.27 0.29
Year 2.2e-14 1 0.89 0.28 -0.048 0.32 0.27 0.89 -0.46 0.26 0.24
Population status month begin 0.16 0.89 1 0.23 0.042 0.23 0.19 1 -0.37 0.26 0.093
Number of relocation to Berlin 0.17 0.28 0.23 1 0.35 0.9 0.92 0.3 -0.099 -0.082 0.15
Number of relocation from Berlin 0.38 -0.048 0.042 0.35 1 -0.096 -0.022 0.039 0.43 -0.28 -0.18
Relocation balance -0.0057 0.32 0.23 0.9 -0.096 1 0.99 0.31 -0.31 0.045 0.24
Population status balance 0.032 0.27 0.19 0.92 -0.022 0.99 1 0.28 -0.24 -0.053 0.22
Population status month end 0.16 0.89 1 0.3 0.039 0.31 0.28 1 -0.38 0.25 0.11
living birth 0.63 -0.46 -0.37 -0.099 0.43 -0.31 -0.24 -0.38 1 0.15 0.042
deaths 0.27 0.26 0.26 -0.082 -0.28 0.045 -0.053 0.25 0.15 1 0.33
Tonnage 0.29 0.24 0.093 0.15 -0.18 0.24 0.22 0.11 0.042 0.33 1
Month
Year
Population status month begin
Number of relocation to Berlin
Number of relocation from Berlin
Relocation balance
Population status balance
Population status month end
living birth
deaths
Tonnage
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

{3} Auswahl gezielter Features



['SPM Sammlung',
'HM',
'BIO',
'Tonnage',
'Shift',
'Tour',
'Tour_1',
'Tour_2'
'Tour_3',
'Tour_4'
'Tour_5',
'Tour_7',
'Tour_8',
'Tour_9',
'Tour_10',
'Tour_12',
'VMWSN',
'VMWSF',
‘deaths’,
'weekday_Monday'
'weekday_Tuesday']

{4} Aggregation der Spalten

```
Date,Month,CW,Year,Yard,Shift,Tour,Tonnage,Type,Tour_Stays,avg_temperature,avg_temperature_1d_before,avg_temperature_2d_before,
02.01.19,1,1,2019,VMF,1,1,"5,59",BIO,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,True,Fa
02.01.19,1,1,2019,VMF,1,4,"3,23",BIO,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,True,Fa
02.01.19,1,1,2019,VMF,1,5,"5,68",BIO,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,True,Fa
02.01.19,1,1,2019,VMF,1,6,"5,48",BIO,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,True,Fa
02.01.19,1,1,2019,VMF,1,7,"7,84",BIO,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,True,Fa
02.01.19,1,1,2019,VMF,1,9,"4,63",BIO,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,True,Fa
02.01.19,1,1,2019,VMF,1,10,"4,38",BIO,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,True,
02.01.19,1,1,2019,VMF,1,12,"9,50",BIO,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,True,
02.01.19,1,1,2019,VMF,1,1,"24,00",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,2,"20,70",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,3,"22,30",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,4,"17,16",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,5,"11,60",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,6,"19,10",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,7,"15,94",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,8,"20,94",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,9,"16,94",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False,
02.01.19,1,1,2019,VMF,1,10,"11,28",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False
02.01.19,1,1,2019,VMF,1,11,"10,54",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False
02.01.19,1,1,2019,VMF,1,12,"22,76",HM,2159841.0,1.8,6.4,5.5,6.2,0.1,2.2,0.4,0.1,True,False,False,False,False,False,False,False
```

Bild 4: Ausschnitt der finalen CSV

{5} Modellentwicklung

1. Aufteilen der Daten in Test-, Trainings- und Validierungsdaten
2. **Trainings** des Modells mit unterschiedlichen Algorithmen
   [Linear, Random_Forest, SGD, DecisionTree]
3. Evaluierung mit geeigneten Metriken
   [RMSE, MAE, R^2]
4. **Plotten** der Ergebnisse
5. Modelloptimierung?

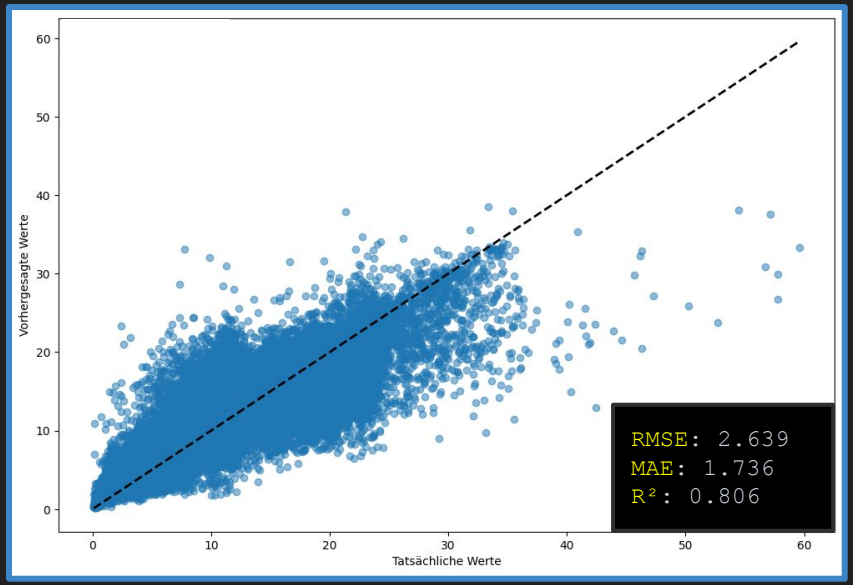
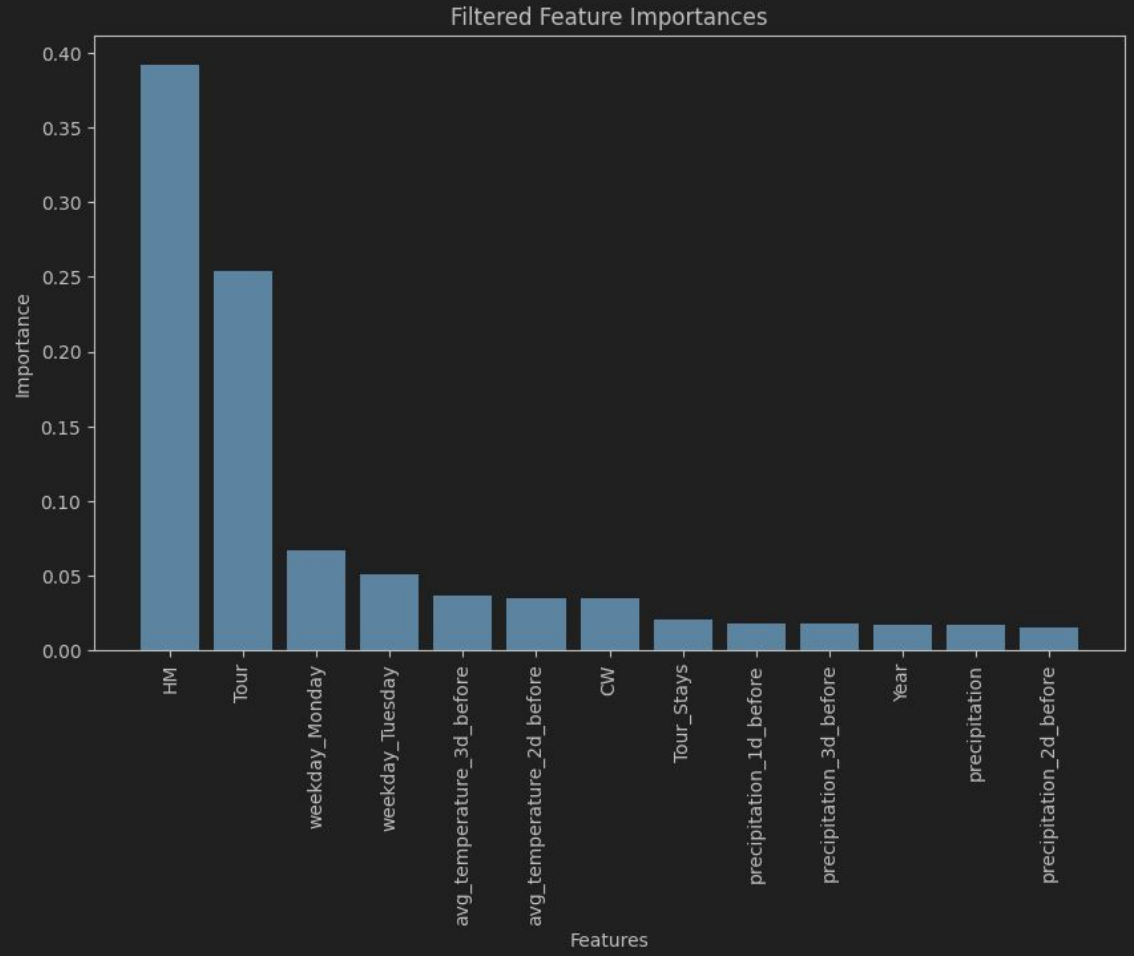


Bild 5: Beste Ergebnisse durch Random Forest Regressor


Filtered Feature Importances
Importance
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
HM
Tour
weekday_Monday
weekday_Tuesday
avg_temperature_3d_before
avg_temperature_2d_before
CW
Tour_Stays
precipitation_1d_before
precipitation_3d_before
Year
precipitation
precipitation_2d_before
Features

{6} Fazit und Verbesserungsmöglichkeiten