Bericht

AWE: KI-gestützte Analyse von Stadtreinigungsdaten zur Optimierung von Einsatzplänen

x

To Uyen Nguyen Thi - 574330

Justin Pallas - 590815

Viet Duc Hoang - 574131

Krist Baliev - 574074

Inhaltsverzeichnis

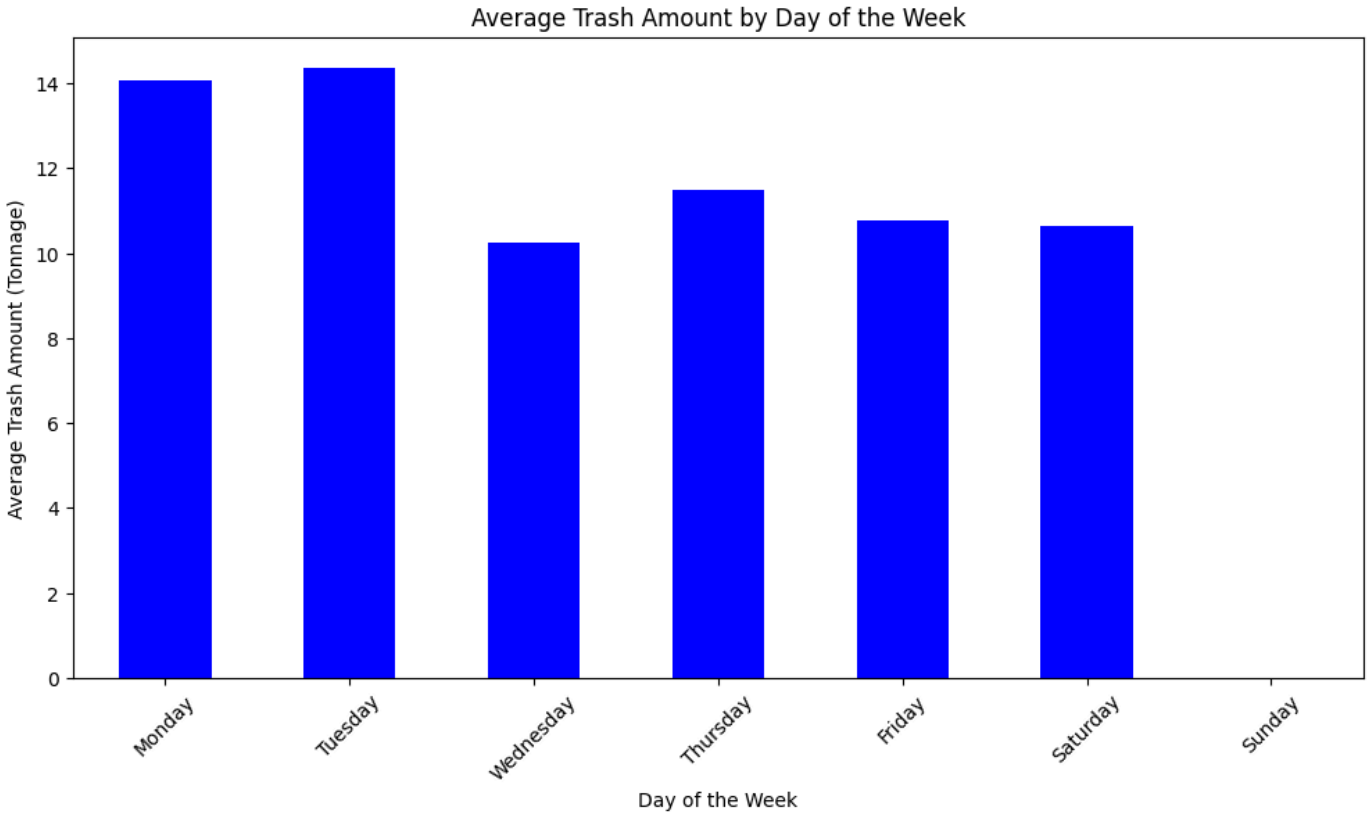
Planung

1. Suche nach externen Datenquellen

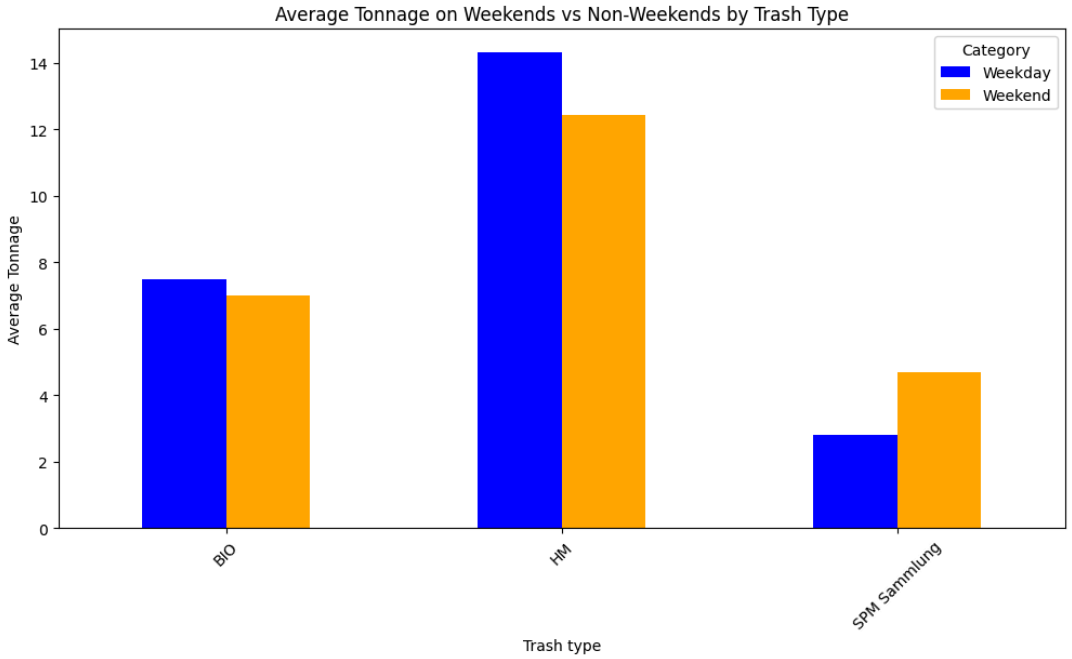
Zu Beginn haben wir uns die von der BSR bereitgestellten Daten genau angeschaut und überlegt, welche äußeren Einflüsse sich auf die Müllmengen auswirken könnten. Anschließend haben wir nach öffentlich verfügbaren Datensätzen gesucht, die uns zusätzliche Features für das Modell liefern könnten. Dabei haben wir uns besonders auf Datensätze in den Bereichen Wetter, Tourismus, Bevölkerung und Ferientagen fokussiert. Zusätzlich haben wir auch Datensätze zur Entwicklung der Verbraucherpreise und zum Umsatz im Einzelhandel mit einbezogen. Bei den herangezogenen Datensätzen ist zu beachten, dass es sich hierbei um zeitlich aufgelöste Daten handelt, meistens nach Monaten. Räumlich aufgelöste Daten, die beispielsweise den Bezirken oder den Postleitzahlen von Berlin zugeordnet werden könnten, wurden nicht mit aufgenommen.

2. Analyse der bereitgestellten BSR-Daten

Um die bereitgestellten Daten besser zu verstehen, haben wir sie mit Hilfe von *matplotlib* visualisiert und untersucht, ob wir auf diesem Weg bereits Auffälligkeiten feststellen können. Dabei haben wir unter anderem festgestellt, dass die Müllmengen abhängig von den Wochentagen stark variieren.

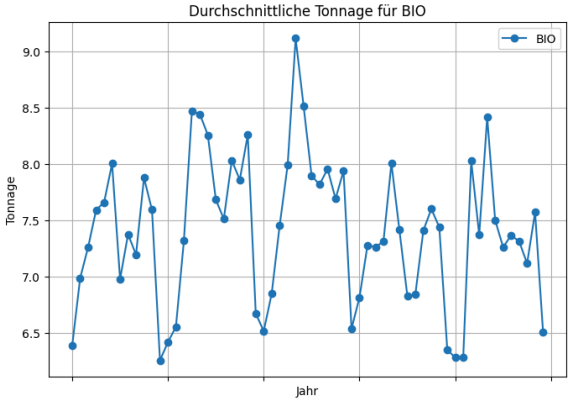


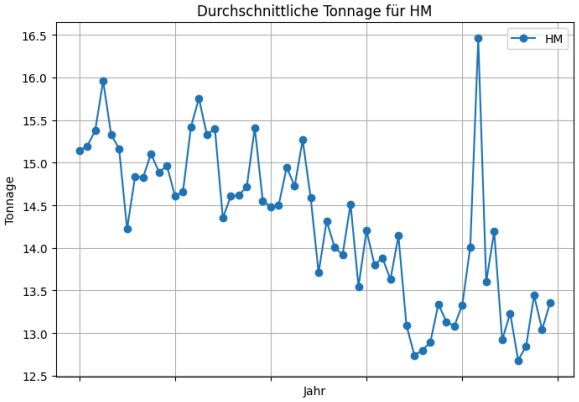
Allerdings scheint der Wochentag nicht nur für die allgemeine Müllmenge wichtig zu sein, sondern beeinflusst auch die Verteilung der verschiedenen Abfallarten. Am Wochenende fällt beispielsweise deutlich mehr Sperrmüll an, als an den restlichen Wochentagen. Allerdings scheint es am Wochenende in der Regel weniger Hausmüll zu geben.

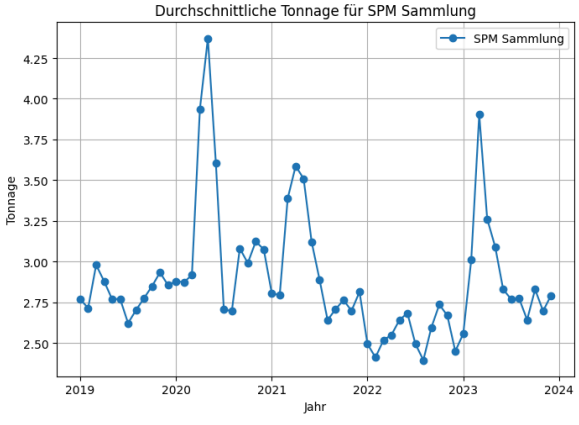


Zusätzlich ist uns aufgefallen, dass die Müllmengen auch je nach Tournummer stark variieren. Wir vermuten, dass das mit der jeweils gefahrenen Route zusammenhängt.

Auch saisonal scheint es Unterschiede zu geben. Im Winter fällt beispielsweise im Schnitt eher weniger Biomüll an, als im Sommer und die durchschnittliche Menge des Hausmülls scheint sogar generell über die Jahre zu fallen.







Datenvorbereitung

1. Import und Bereinigung

Nachdem wir die Daten herausgesucht hatten, haben wir diese zunächst mit *python* eingelesen und mithilfe der *pandas* library zu einem Dataframe zusammengesetzt. Anschließend haben wir die Daten bereinigt und dabei fehlende Werte aufgefüllt (z.B. mit dem Durchschnittswert der jeweiligen Zeile).
Manche der externen Daten (wie der Verbraucherpreisindex) lagen uns nur in einer monatlichen Auflösung vor, sodass wir diese künstlich zu täglichen Werten aufgefüllt haben, indem wir beispielsweise durch die Anzahl der Tage des jeweiligen Monats dividiert haben.

2. Zusammenführen der externen Daten und der BSR-Daten

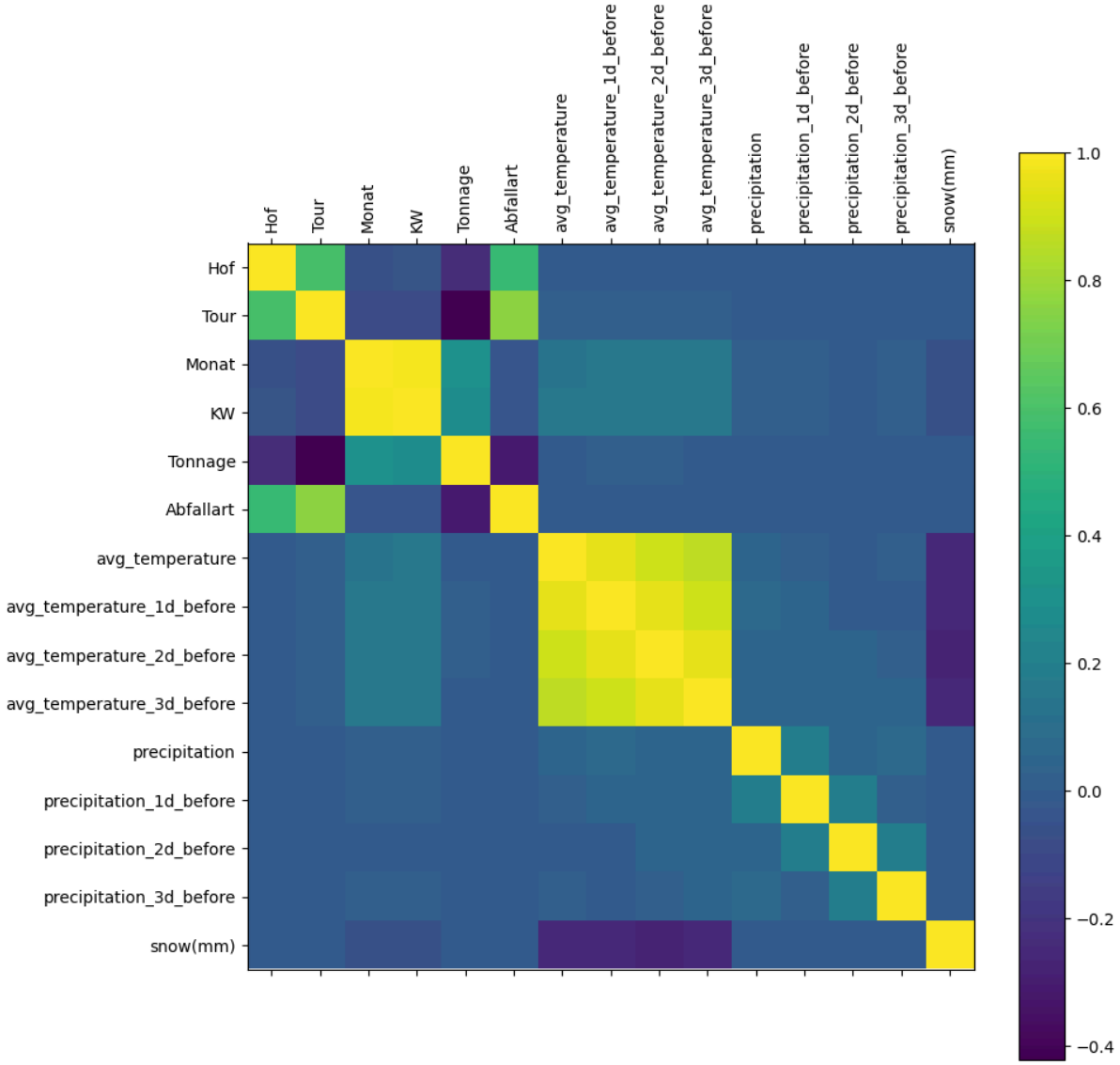
Um die Daten zusammenzuführen, haben wir die Datums-Spalte als Index verwendet und jeder Zeile der BSR-Daten die entsprechenden Informationen zu dem jeweiligen Datum aus den externen Daten hinzugefügt. Die zusammengeführten Daten haben wir anschließend im CSV-Format gespeichert und mit Git versioniert.

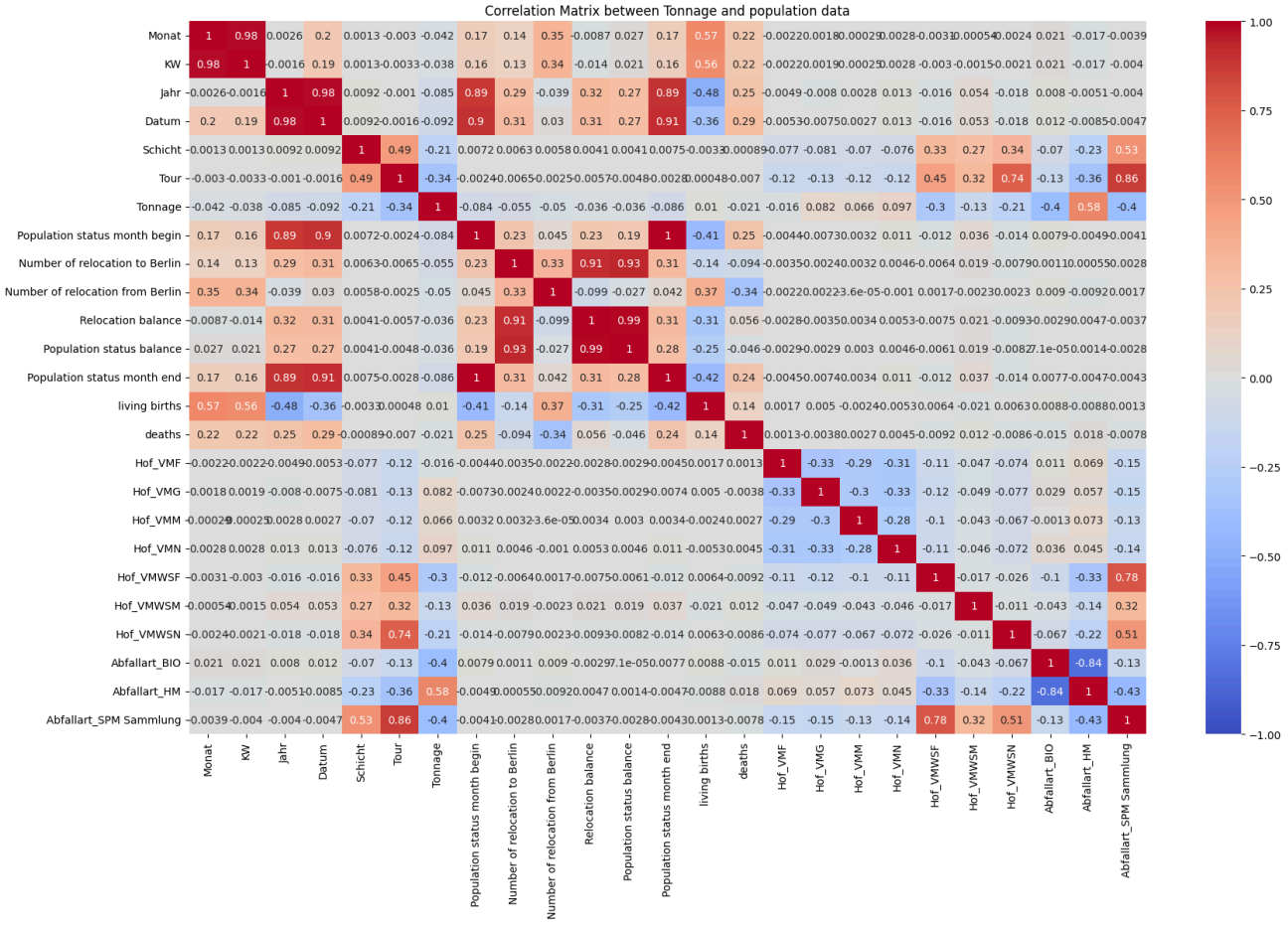
3. Feature Engineering

Bei der Aufbereitung der Daten haben wir verschiedene Feature Engineering Techniken verwendet. Die Wochentage haben wir beispielsweise mittels One-Hot Encoding gespeichert, da jedes Datum immer nur einen Wochentag haben kann und so jeder Wochentag numerisch einheitlich dargestellt wird.
Für die Wetterdaten sind lag-Features erstellt worden, womit zusätzliche Spalten mit einem zeitlichen Verzug von ein bis drei Tagen erstellt wurden.
Um alle Features gleich stark zu gewichten, haben wir die Werte der einzelnen Features zusätzlich normalisiert und auf einen Wertebereich zwischen 0 und 1 gebracht.

4. Finden von Korrelationen zwischen externen und internen Daten

Nachdem wir die Daten zusammengeführt haben, haben wir nach Korrelationen der einzelnen Features mit der Tonnage gesucht. Dabei haben wir überwiegend mit Korrelationsmatrizen gearbeitet. Allerdings haben wir hier festgestellt, dass die meisten der von uns extern hinzugefügten Features nur eine sehr minimale bis gar keine Korrelation mit der Müllmenge haben.



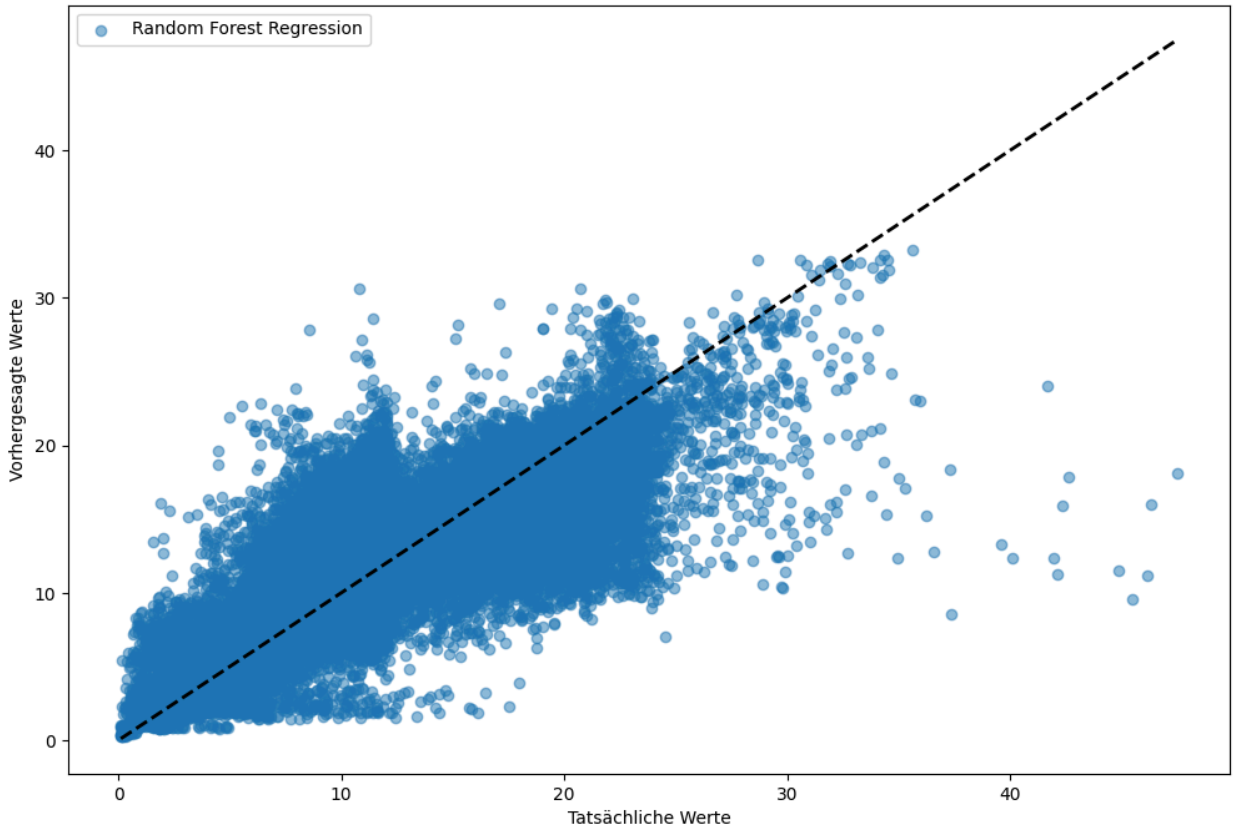


Da wir allerdings bereits viel Zeit in die Datenbeschaffung und -aufbereitung investiert hatten, haben wir uns dazu entschieden, einige der Features dennoch zunächst einmal in das Modell mit aufzunehmen und auszuprobieren, ob es nicht eventuell doch zu leicht besseren Ergebnissen führt.

Modellentwicklung

Zu Beginn haben wir unsere Daten in Test-, Trainings- und Validierungsdaten aufgeteilt und das Modell mit verschiedenen Algorithmen (Linear Regression, SGD, DecisionTree, RandomForest) auf 213 Features (größtenteils aus dem One-Hot-Encoding für die Touren) trainiert. Dabei haben wir 75% der Daten für das Training benutzt. Die Daten wurden nicht geshuffelt, also chronologisch aufgeteilt. Für das erste nicht-optimierte Modell kamen folgende Metriken raus:

RMSE: 3.0943573155634363,
MAE: 2.1267839766733,
R^2: 0.7075959661027306



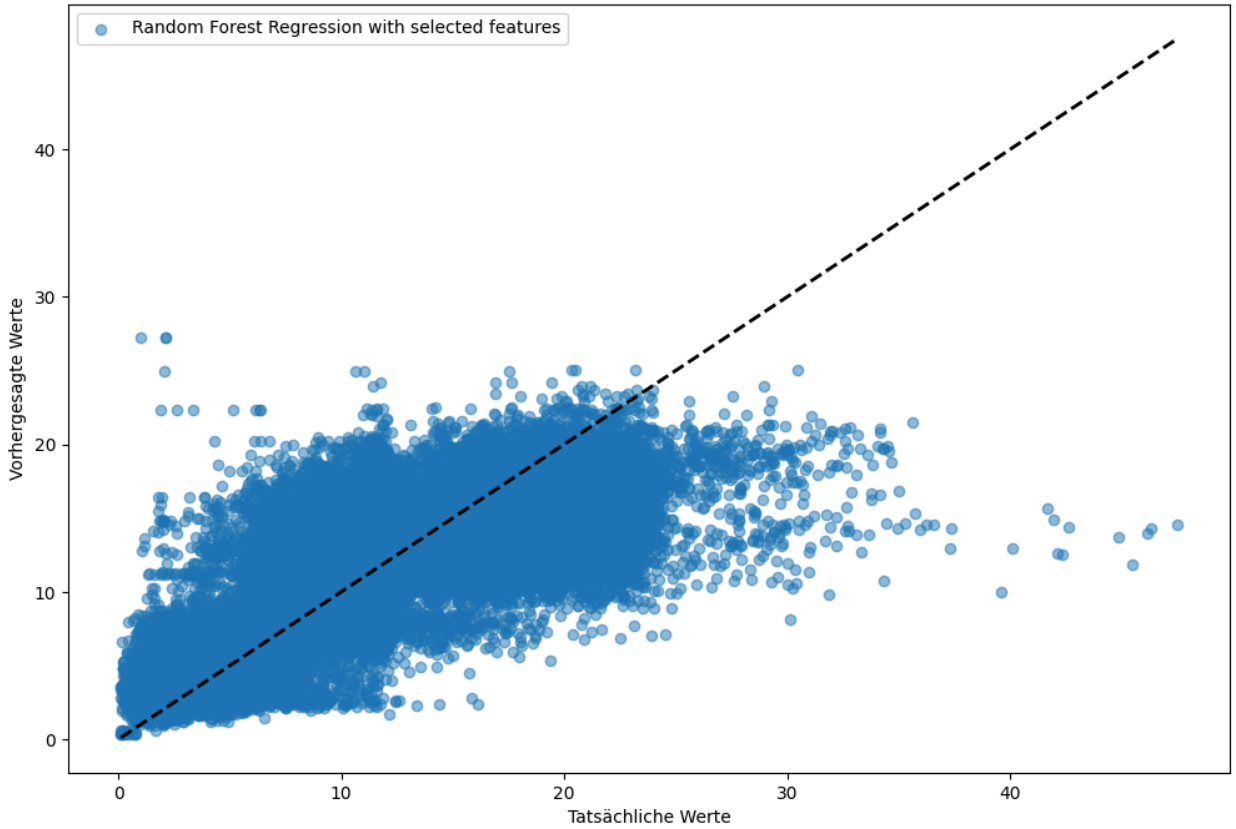
Zusätzlich haben wir auch ausprobiert, die Features, bei denen wir vorher keine nennenswerten Korrelationen feststellen konnten, aus dem Modell zu entfernen.

Von Relevanz waren dann daher nur noch folgende Features:

```
selected_columns = [
'Month',
'CW',
'Year',
'SPM Sammlung',
'Tonnage',
'Tour_7',
'Tour_8',
'Tour_3',
'Shift',
'Tour_12',
'VMWSN',
'Tour',
'Tour_10',
'Tour_1',
'Tour_2',
'VMWSF',
'weekday_Tuesday',
'HM',
'BIO',
'Tour_4',
'Tour_5',
'Tour_9',
'weekday_Monday',
'Tour_Stays',
'avg_temperature_2d_before','avg_temperature_3d_before','precipitation','precipitation
_1d_before','precipitation_2d_before','precipitation_3d_before', 'is_holiday']
```

Hierbei wurde das Modell tatsächlich etwas ungenauer.

Metriken aus dem Random Forest Regressor mit ausgewählten Features:
RMSE: 3.827666927767836,
MAE: 2.8921898465705778,
R^2: 0.5525848189940457



Bei der Auswahl dieser Features ist aber auch zu sagen, dass aufgrund der Zeit Features nicht wirklich "wissenschaftlich fundiert" heraus selektiert wurden. Es war der Wunsch, nur Features mit einer Korrelation von größer als 0.3 aufzunehmen. Manchmal wurden aber auch kleine Werte akzeptiert.

Für die Optimierung war noch geplant, mit Hilfe von GridSearchCV und einem Parameter-Grid das Modell auf verschiedene Parameter zu trainieren, jedoch hat die Zeit nicht mehr dafür ausgereicht, da das Training unseres Modells mit RFG alleine 3 Minuten gedauert hat, und mehrere Iterationen nicht beendet werden konnten.

Fazit

Rückblickend haben wir vermutlich etwas zu viel Zeit mit der Beschaffung geeigneter externer Datensätze verbracht. Wir hätten mehr Wert darauf legen sollen, täglich aufgelöste Datensätze zu finden und diese mit einzubeziehen. Bei monatlich aufgelösten Daten sollte man nach Methoden Ausschau halten, die eine sinnvollere Verteilung des Monatswerts hergeben, anstatt nur mit der Anzahl der Tage des Monats zu dividieren, was eine zu einfache Annahme ist.

Aufgrund der Datenakquise und der Datenvorbereitung hatten wir nur wenig Zeit, um das Modell zu trainieren und konnten hier nicht viel ausprobieren. Wir gehen davon aus, dass die Anzahl der aufgenommenen Features einfach zu hoch war, weshalb hier nochmal eine ausgiebige Anwendung an Feature Selection von Nöten wäre.

Insgesamt konnten wir allerdings auch trotz der kurzen Zeit viele nützliche Informationen aus den bereitgestellten Daten gewinnen und erste Modelle entwickeln.