Tonnage-Prediction-AI

HTW - AWE KI Ana
KI-gestützte Analyse von Stadtreinigungsdaten
zur Optimierung von Einsatzplänen

Jannes, Leonardo, Sabine, Nicole

Ziel des Projekts

Ziel des Projekts war es, die bereitgestellten BSR-Daten zu explorieren, diese gegebenenfalls zu bereinigen, sowie zusätzliche relevante Features aus externen Datensätzen zu integrieren. Daraufhin sollten mehrere Modelle für die Vorhersage neuer Touren bezüglich der Tonnage entwickelt, optimiert und miteinander verglichen werden.

Inhalt

Beschreibung und Analyse der BSR-Daten

Der Ausgangspunkt des Projektes war die gegebene csv-Datei, die historische Daten über Touren der BSR im Zeitraum von 2019 bis 2024 enthält. Dabei enthält der Datensatz folgende Features:

- **Monat**: Monat der Abfallanlieferung.
- **KW (Kalenderwoche)**: Kalenderwoche der Abfallanlieferung.
- **Datum**: Datum der BSR-Tour (Samstags und Sonntags wird in der Regel kein Abfall gesammelt, außer es handelt sich um einen Nachladetag nach Feiertagen oder durch Krankheits-/Fahrzeugausfälle).
- **Hof**: Zuordnung des Hofes zur Tour (VMF = Hof Forckenbeckstraße, VMG = Hof Gradestraße, etc.).
- **Schicht**: Schicht der Abholung
- **Tour**: Nummer der Auslieferungstour (jede Tour hat eine täglich rotierende Route, die sich alle zwei Wochen wiederholt; Sperrmülltouren arbeiten auftragsbezogen ohne feste Touren).
- **Tonnage**: Gelieferte Abfallmenge in Tonnen.
- **Abfallart**: Art des Abfalls (BIO = Biomüll, HM = Hausmüll, SPM = Sperrmüll).

Auffällig war zu Beginn, dass der Datensatz sehr viele NaN-Zeilen zum Schluss enthielt, die wir direkt gedroppt haben, was den Datensatz von 388.185 auf 381.661 Zeilen reduzierte. Zudem gab es viele Daten-Typen, die für typische ML-Modelle unpraktisch sind und die wir deshalb anpassen mussten. Wir haben zum Beispiel die Spalten Hof und Abfallart per One-Hot-Encoding kodiert und die ursprünglichen Spalten gedroppt. Auch der Datentyp des Datums musste per cast auf den date-Typ von pandas angepasst werden. Zudem war auch

die Tonnage als object gegeben und musste erst in float64 umgewandelt werden. Darüber hinaus haben wir nur ein echtes Duplikat gefunden und dieses entfernt.

Bei der explorativen Datenanalyse ist uns zunächst aufgefallen, dass die Tonnage an den einzelnen Wochentagen sehr schwankt, wobei dieser Wert am Mittwoch am niedrigsten und am Montag am höchsten ist. Außerdem scheinen Hausmüll-Touren eher höhere Tonnagen zu produzieren, da die Werte miteinander korrelieren. Es gibt zudem auch einen Trend, dass von Jahr zu Jahr tendenziell die Anzahl der Touren pro Woche steigt. Im Frühling 2023 haben wir in der Tonnage, sowie in der Anzahl der Touren starke Ausreißer festgestellt, was wir durch eine kurze Recherche auf BSR-Streiks in diesem Zeitraum zurückführen konnten.

Externe Datensätze

Da die gegebenen Features des Datensatzes relativ wenig mit der Zielvariable Tonnage korreliert haben, entschieden wir uns dazu, nach weiteren Datensätzen im Internet zu suchen, die eventuell mehr Information bereitstellen für das spätere Training. Die resultierenden Datensätze sind im folgenden kurz zusammengefasst.

1. **Tägliche Wetterdaten für Berlin (2019-2023)**
   Quelle: Free-Weather-API
   Verwendete Merkmale: (in Abhängigkeit von Datum)
   - Temperatur [C]
   - Regenmenge [mm]
   - Schneefall [mm]
   - Sonnenstunden [s]
2. **Monatliche Inflationsdaten für Nahrungsmittelgruppen (2020-2023)**
   Quelle: Statistisches Bundesamt
   Inflationsdaten gemessen an der 100%-Marke für folgende Lebensmittelkategorien:
   - Brot
   - Fleisch
   - Fisch
   - Molkereiprodukte und Eier
   - Speisefette und -öle
   - Obst
   - Gemüse
   - Zucker, Marmelade, Honig
   - Fertiggerichte
   - Kaffee/Tee
   - Wasser/Saft
3. **Wahlumfragen für Berlin (2019-2023)**
   Quelle: https://www.wahlumfragen.org/berlin/
   Prozentangaben für die Parteien:
   - SPD
   - CDU
   - Grüne
   - Linke

    - AFD
    - FDP
    - Sonstige
4. **Feiertage und Schulferien in Berlin (2019-2023)**
   Quellen:
    - Feiertage: https://nager.date.at/
    - Schulferien: Berliner Schulkalender

Da alle verwendeten Datensätze aus vertrauenswürdigen Quellen stammen, gehen wir von ihrer Richtigkeit aus. Die Daten wurden in CSV-Dateien konvertiert und ggf. nur die relevanten Informationen extrahiert, um sie für das Feature Engineering und Modelltraining nutzbar zu machen.

Ein wichtiger Punkt ist der Inflationsdatensatz, der für das Jahr 2019 keine Werte enthält. Für das Training wurden daher nur die Jahre 2020-2023 berücksichtigt, oder wir haben die Inflationsdaten aus dem Datensatz entfernt, wenn der gesamte Zeitraum einbezogen wurde.

Feature-Engineering

Feature Engineering ist der Prozess, bei dem Rohdaten in ein Format umgewandelt werden, das besser für Machine-Learning-Modelle geeignet ist. Es umfasst die Auswahl, Transformation und Erstellung von Merkmalen (Features), um die Leistung des Modells zu verbessern.

Wir haben uns dabei vor allem auf zeitliche Transformationen wie lag- oder rolling-features konzentriert, da wir einen starken Zusammenhang zu den Werten von den Tagen davor vermuteten. Zum Beispiel haben wir das Feature Tonnage mehrmals als lag-Feature transformiert, so dass jede Spalte die Informationen enthält wie die Tonnage die letzten acht (also jeweils lag 1-8) Tage war. Zudem haben wir bei allen Wetterfeatures die Werte der letzten 3 Tage als rolling-Feature gemittelt hinzugefügt.

Zuletzt haben wir ein Feature erstellt, das für jede Tour feststellt, wie viele Tage zuvor die BSR nicht gearbeitet hat.

Nach der Integration aller relevanten Features besteht unser finaler Datensatz nun aus insgesamt **91 Spalten**. Die Spalten umfassen sowohl die originalen BSR-Daten als auch die extrahierten und berechneten Merkmale, die aus den externen Datensätzen stammen.

Modelltraining

Für das Modelltraining haben wir verschiedene Ansätze verwendet:

1. **Lineare Regression**
2. **Random Forest**
3. **XGBoost**

Die Modelle wurden sowohl auf dem originalen BSR- als auch auf dem finalen Datensatz trainiert und mit verschiedenen Hyperparametern optimiert. Anschließend wurden die Ergebnisse der Modelle verglichen, um das am besten geeignete Modell für die Vorhersage der Abfallmengen zu bestimmen.

Die Daten wurden zuvor in Trainings-, Test- und Validierungsdaten aufgeteilt, wobei die zeitliche Struktur der Daten beachtet wurde, um Datenlecks zu vermeiden. Die Evaluierung erfolgte mit den Metriken MAE, RMSE und R^2-Score. Die Ergebnisse sind in den folgenden Tabellen zusammengefasst:

Lineare Regression

Nach der Standardisierung der entsprechenden Features wurde das Modell trainiert und evaluiert. Eine Log-Transformation der Zielvariable wurde angewendet. Zur Optimierung des Modells wurden auch Ridge und Lasso Regression durchgeführt. Abschließend erfolgte die Evaluation auf den Testdaten mit folgenden Metriken:

Metrik (Testdaten)	Original BSR Datensatz	Finaler Datensatz
Mean Absolute Error (MAE)	3.88	3.21
Root Mean Squared Error (RMSE)	4.75	4.38
Mean Squared Error (MSE)	22.53	19.17
R^2-Score	0.14	0.497

Das detaillierte Vorgehen zur linearen Regression, einschließlich Datenvorbereitung, Modelltraining und Evaluation, sind im Notebook “lineare_regression_all_data” und “lineare_regression_bsr_only” dokumentiert.

Random Forest

Ein Random Forest Regressor ist ein Ensemble-Lernverfahren, das mehrere Entscheidungsbäume trainiert und deren Vorhersagen mittelt, um robuste und genaue Ergebnisse zu erzielen. Jeder Baum wird auf einem zufälligen Teil der Trainingsdaten mit zufällig ausgewählten Features trainiert, wodurch Overfitting reduziert und die Generalisierungsfähigkeit verbessert wird.

Für das Training mit dem RFG haben wir zunächst den Datensatz mithilfe eines TimeSeriesSplit in 5 Blöcke aufgeteilt. Als nächstes haben wir den RFR initialisiert und ein Parameter Grid mit typischen Hyperparametern erstellt. Das Modell wurde daraufhin aus Effizienz mit Hilfe einer HalfingRandomSearch mit dem Scoring “negative mean squared error” trainiert. Das beste Modell hatte einen MSE von 15.68 auf den Trainingsdaten.

XGBoost

XGBoost Regression haben wir zum einen auf dem originalen BSR Datensatz ohne Hinzufügen weiterer Features angewandt und auf unserem finalen Datensatz mit den beschriebenen Features. Um die Ergebnisse zu verbessern, wurde im Original Datensatz eine Feature Selection durchgeführt und im finalen Datensatz zusätzlich noch GridSearch, um die besten Parameter für das Training zu finden. Die Ergebnisse sind in der Tabelle zu sehen. In den Notebooks “training_XGBOOST” und “training_XGBOOST_BSRonly” finden sich teilweise noch nähere Erläuterungen dazu.

Metrik	Original Datensatz - Vor Tuning	Original Datensatz - Nach Tuning	Finaler Datensatz - Vor Tuning	Finaler Datensatz - Nach Tuning
Mean Absolute Error (MAE)	3.61	3.55	2.75	2.69
Root Mean Squared Error (RMSE)	4.52	4.33	3.73	3.66
R^2-Score	0.2256	0.2885	0.6335	0.6478
Trainingszeit	0.43 sek	0.52 sek	1.09 sek	1.56 sek
Validierungs(fehler) MAE/RMSE R^2	3.25/4.15 0.4912	3.31/4.11 0.5021	2.44/3.21 0.6960	2.32/3.08 0.7193

Reflexion und Ausblick

Die hinzugefügten Features aus den Datensätzen hatten insgesamt nur einen geringen Einfluss auf die Vorhersagegenauigkeit. Besonders relevant für das Modelltraining war jedoch das Merkmal „Tour“ sowie das von uns eingeführte Feature “Tonnage_lag_2”. Die Einbeziehung weiterer relevanter Features wie zum Beispiel verschiedene Varianten von lag- und rolling-features von Tonnage scheinen vielversprechender zu sein, als externe Daten.

Eine Optimierung wäre möglich, insbesondere durch eine detailliertere Gruppierung der Daten nach Tagen und Höfen. Dies könnte die Grundlage für eine tagesgenaue Vorhersage schaffen und die Modellgenauigkeit weiter verbessern.

Für das weitere Vorgehen bietet sich eine Verbesserung der **Datenaufbereitung** an, um die Modellleistung zu steigern. Zudem könnte eine **erweiterte Feature Selection** sowie die Integration zusätzlicher **Datensätze** hilfreich sein. Ein tieferes Verständnis der Modelle und des Trainingsprozesses ist erforderlich, um die Leistung zu optimieren. Darüber hinaus könnten weitere **Modellen** angewendet werden, um den besten Ansatz zu finden.