Forecasting wastewater discharges with Machine Learning

Capstone Project
Professional Certificate in Data Analytics
Imperial College London

Marcin Lenarcik
September 2025

Executive summary

Accurate predictions of wastewater discharges could support a proactive management of treatment sites and help to optimally utilise their capacity alongside long-term infrastructure investment. The purpose of this project is to assess the potential of Machine Learning (ML) methods for making such predictions.

Initial applications of ML models show promising results, indicating that these models can be used to predict potential discharges based on rainfall forecasts.

Analysis of wastewater discharges and rainfall data suggest that the two are correlated with larger discharges typically occurring after extended periods of rain, particularly in winter.

However, further analytical work is required, as the current prediction quality is not yet at satisfactory level. Optimisation and exploration of different ML approaches is recommended, along with expanding the dataset to include all wastewater treatment sites.

Aerial view of wastewater treatment site next to river and lake in England.

Introduction

Wastewater treatment in the UK

- Wastewater (sewage) is a mix of domestic and industrial wastewater, and rain collected from roads.
- Each day, over 11 billion litres of wastewater are collected by 350,000 km of sewers.
- 9,000 sewage sites treat wastewater before it is discharged to rivers, lakes and the sea.
- Treatment of wastewater is essential for protecting the environment.

Wastewater discharges

- Heavy rainfall can exceed capacity available at treatment sites and lead to discharge of wastewater (untreated / partially treated).
- These discharges are strictly regulated and closely monitored by event duration monitors.
- Data about each discharge is shared by water companies and available from the Environment Agency.

Source: earthwatch.org.uk

Introduction

Impacts of wastewater discharges

- In 2024, over 450 thousand discharges were recorded with total duration of over 3.6 million hours.
- This is a major environmental problem and serious issue for people using rivers, lakes and the sea.
- Building sufficient capacity to deal with future events requires major investment in infrastructure and many years to deliver it.
- Optimal utilisation of capacity available today could be strategically strengthened with proactive use of data-driven analytics and forecasting methods.

Can data help?

- Data availability and the physical link between rainfall and discharge motivate the assessment of data-driven methods.
- In this report we first analyse to what extent the occurrence of discharge events depends on rainfall during and on the days prior to the discharge.
- Then, we explore several ML methods to assess their suitability for accurate forecasting of future discharge events.
- Lastly, we discuss the results and suggest next steps.

Source: DEFRA, environment.data.gov.uk/dataset

Methodology

Data sources

- In this report a sample of all data (population) was analysed.
- Data for the largest water company (Thames Water) was selected and 4 sites with the highest number of discharges were analysed.
- A full and complete data set is available since 01.04.2022.

1) Wastewater discharges

- Data sources:
  - environment.data.gov.uk/dataset
  - https://github.com/AlexLipp/sewage-map
- Data structure:
  - Total daily rainfall data on 25km grid.
  - 4,445,376 rows x 11 columns.
  - Units: minutes of discharge.

2) Rainfall

- Data sources:
  - Met Office HadUK-Grid
  - https://catalogue.ceda.ac.uk
- Data structure:
  - Each discharge event recorded with site name, start and end date / time (data not used: permit number, river catchment, X Y coordinates).
  - 59,289 rows x 9 columns.
  - Units: millimeters of rain.

Methodology

Data preprocessing - discharge events

- Source data in json format
- Start- and end- DateTime in Unix timestamp in milliseconds converted to standard datetime format.
- Discharge events for a selected sewage treatment site filtered by LocationName.
- Event records range from a few minutes to several days for large, continuous discharges.
- Event data transformed into single daily value by summing up the duration of events on the same day (and location) and / or separating multiday events into each day (max. 1440 minutes per day).

```python
# Define function for discharge data transfromation
def observations_to_daily_minutes(
    df,
    start_col="StartDateTime",
    end_col="StopDateTime",
    tz=None,
    start_date=None,
    end_date=None,
    day_col_name="date",
):
```

```python
daily = observations_to_daily_minutes(
    df_d,
    start_col="StartDateTime",
    end_col="StopDateTime",
    start_date="2022-01-01",
    end_date="2024-12-31"
)

# Add to main dataframe
df['discharge'] = df['date'].map(daily.set_index('date')['minutes'])
```

```python
df.head()
```

	date	location	rainfall	discharge
0	2022-04-01	Newbury	0	0.0
1	2022-04-02	Newbury	0	0.0
2	2022-04-03	Newbury	0	0.0
3	2022-04-04	Newbury	0	0.0
4	2022-04-05	Newbury	0	0.0

Methodology

Data preprocessing - rainfall

- Source data in NetCDF format
- One entry per day kept (total daily rainfall between 9am and 9am the next day) by filtering on category ‘bnds’ == 0
- Date (‘time’), rainfall, latitude and longitude kept for each 25km grid (other columns dropped).
- Rainfall per day for a selected sewage treatment site is filltered and copied to the main dataframe by specifying latitude and longitude of the site. The nearest 25km grid box is used.

```
df_r.head()
```

	time	rainfall	latitude	longitude
0	2022-01-01	0.0	48.833482	-10.009903
2	2022-01-01	0.0	48.856586	-9.672318
4	2022-01-01	0.0	48.878707	-9.334313
6	2022-01-01	0.0	48.899844	-8.995904
8	2022-01-01	0.0	48.919993	-8.657109

```
# Filltering for rows for selected locaiton
# Find nearest grid lat / lon
df_r['distance'] = ((df_r["latitude"] - lat)**2 + (df_r["longitude"] - lon)**2)**0.5
df_nearest = df_r.loc[df_r['distance'].idxmin()]
lat_ = df_nearest['latitude']
lon_ = df_nearest['longitude']
# Fillter only rows for selected location
df_r = df_r[(df_r['latitude'] == lat_) & (df_r['longitude'] == lon_)]

# Add to main dataframe
df['rainfall'] = df['date'].map(df_r.set_index('time')['rainfall'])
```

```
df.head()
```

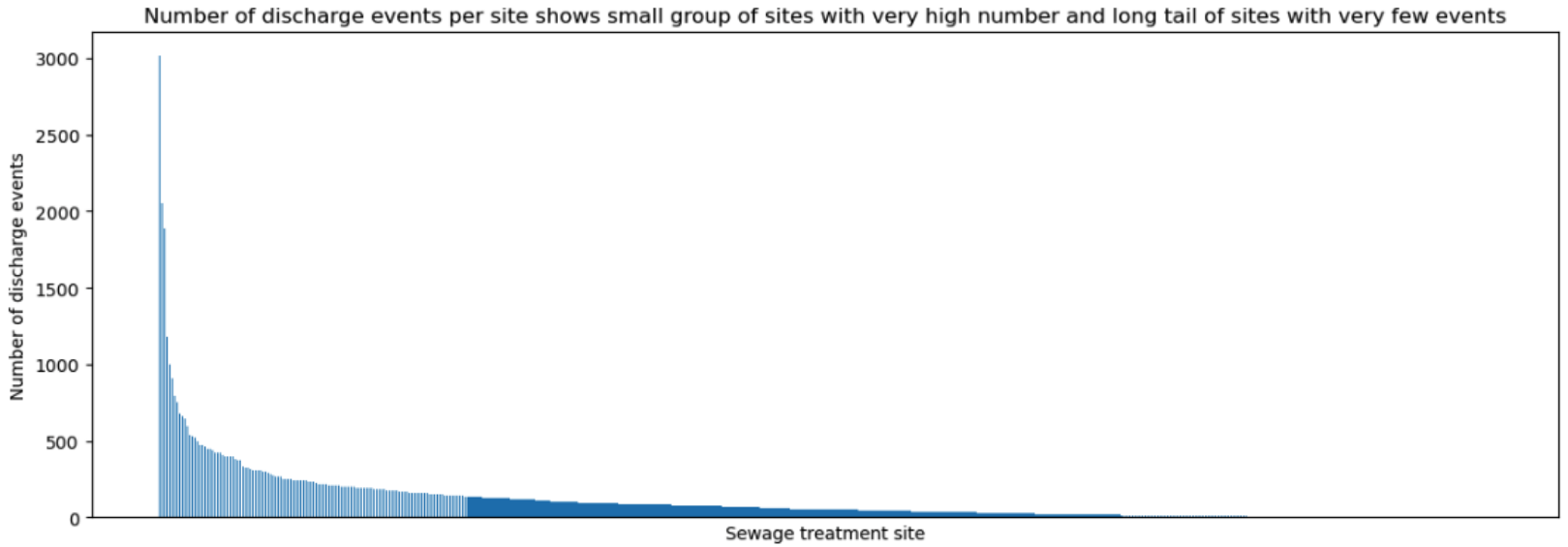
	date	location	rainfall	discharge
0	2022-04-01	Newbury	0.29	0
1	2022-04-02	Newbury	0.12	0
2	2022-04-03	Newbury	4.14	0
3	2022-04-04	Newbury	0.21	0
4	2022-04-05	Newbury	0.88	0

Methodology

Exploratory Data Analysis

- Number of discharge events per site is not evenly distributed.
- Top 10 sites with highest number of discharges.

LocationName	
Windsor	3017
Chinnor	2055
Marlborough	1884
St Stephens Hall (Little London)	1182
Newbury	1000
South Moreton	905
Bampton	793
Farnborough (Warks)	753
Nightingale Lane CSO	679
Willingale	660

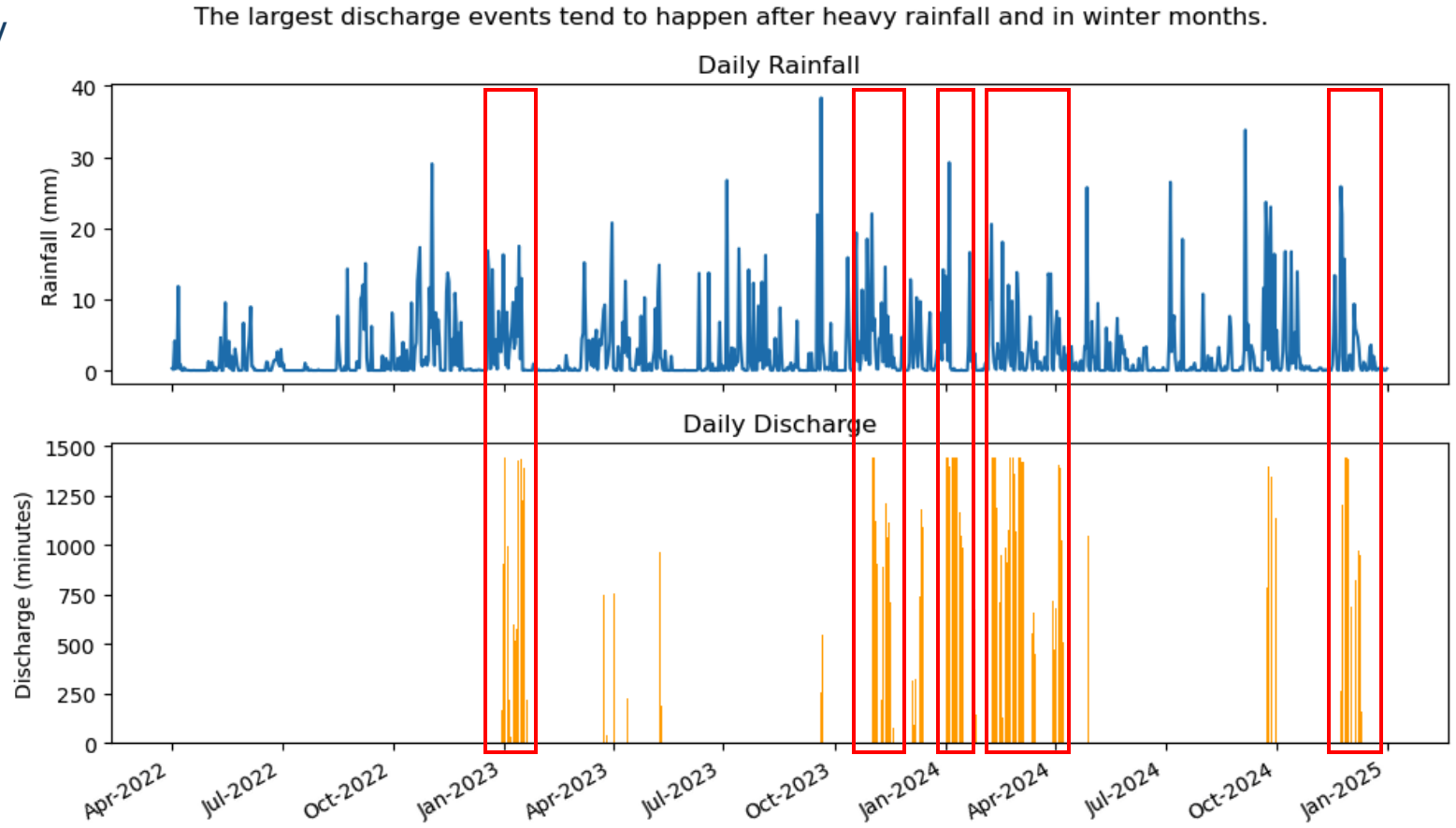


- Data from the top 5 sites has been analysed.

Methodology

Exploratory Data Analysis

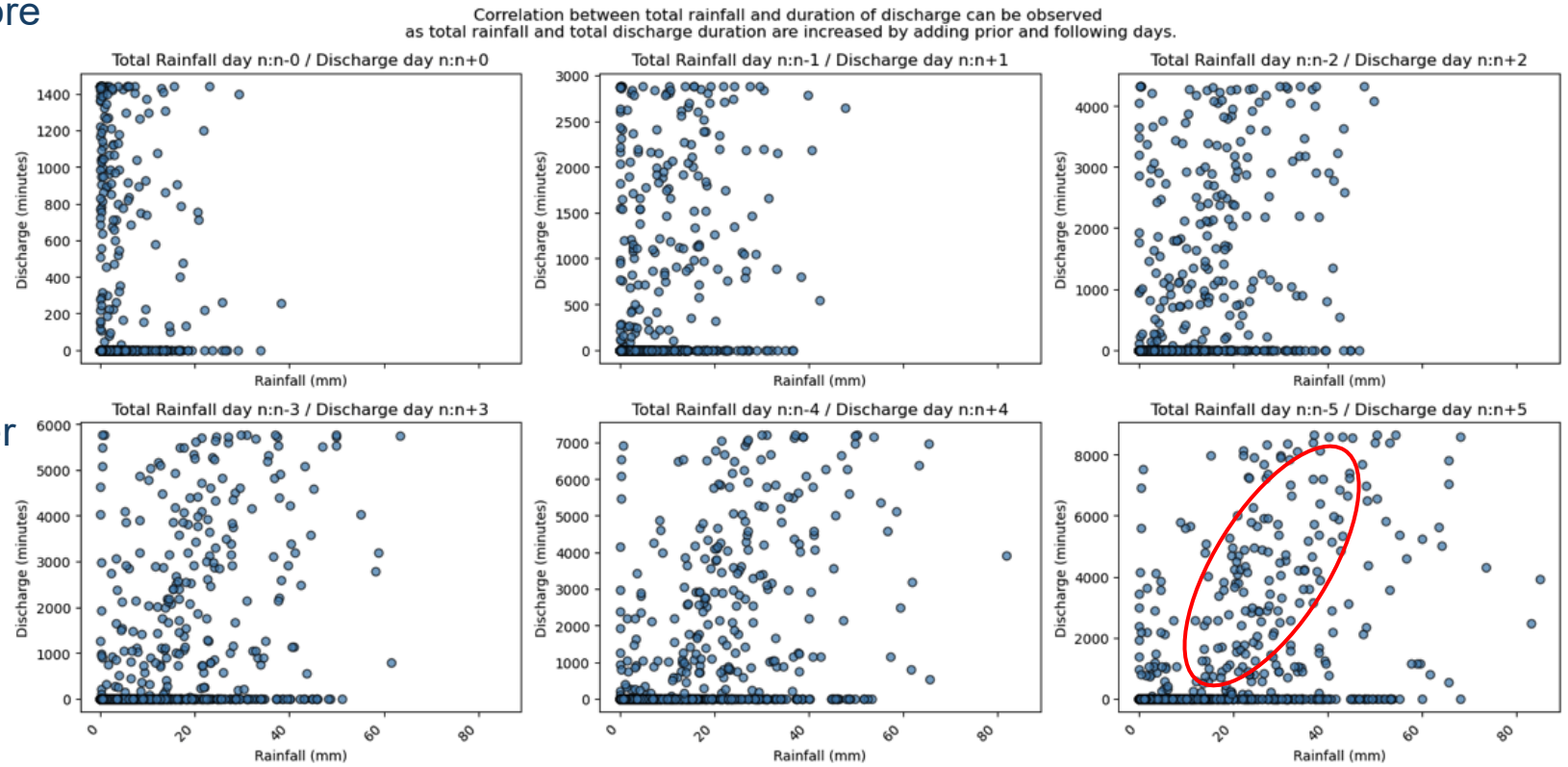
- The largest discharge events tend to:
  - happen after higher daily rainfalls (especially over multiple days)
  - Take place in the winter months.
  - Data for Newbury.



Methodology

Exploratory Data Analysis

- To explore the impact of prolonged rainfall, additional variables were calculated.
- Total rainfall over multiple days:
  - n + (n-1) day n and day before
  - n + (n-1) + (n-2)
  - …
  - n + … + (n-5)
- Discharge duration was similarly processed but for following days:
  - n + (n+1) day n and day after
  - n + (n+1) + (n+2)
  - …
  - n + … + (n+5)
  - Data for Newbury.

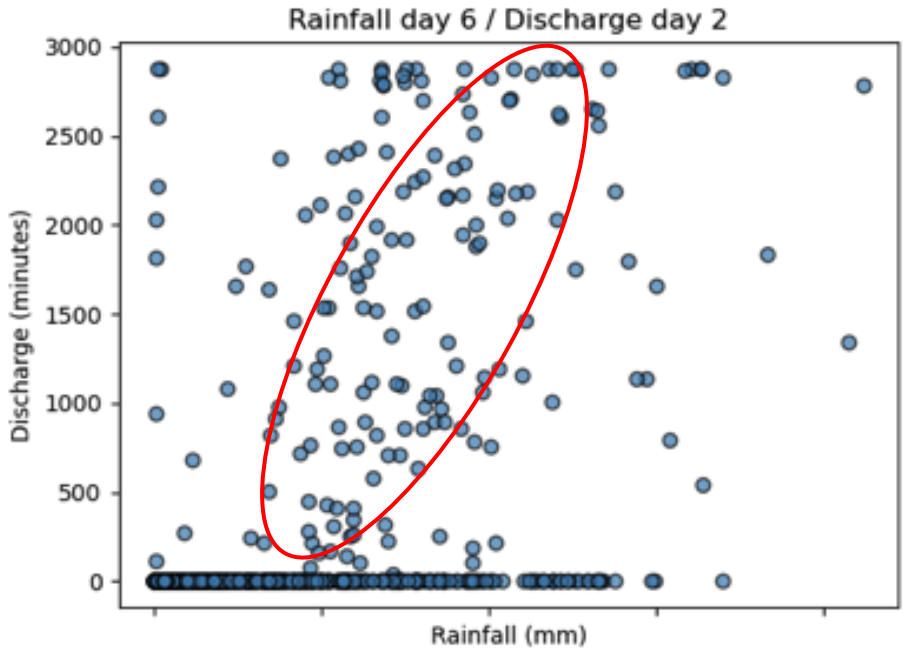


Methodology

Exploratory Data Analysis

- Correlation matrix shows the highest correlation (0.50) between total rainfall over the prior 6 days and total discharge duration in the following 2 and 3 days.
  - Data for Newbury.

	r_d_n-0	r_d_n-1	r_d_n-2	r_d_n-3	r_d_n-4	r_d_n-5
d_d_n+0	0.13	0.30	0.38	0.42	0.46	0.47
d_d_n+1	0.25	0.38	0.43	0.47	0.49	0.50
d_d_n+2	0.28	0.38	0.44	0.47	0.49	0.50
d_d_n+3	0.28	0.39	0.44	0.47	0.48	0.49
d_d_n+4	0.29	0.39	0.44	0.46	0.48	0.48
d_d_n+5	0.29	0.38	0.43	0.45	0.46	0.47



Methodology

Exploratory Data Analysis

- Considering insight from EDA, additional variables added to the dataset:
  - ‘days_since_rain’ – number of days since last recorded rainfall > 25th quantile of rainfall for that location
  - ‘month’ – calendar month recorded as number (1 – January, 2 – February, …, 12 – December)
  - ‘season’ – season recorded as a number (1 – Spring, 2 – Summer, 3 – Autumn, 4 – Winter)
  - ‘target’ – Boolean variable where 1 – day with discharge, 0 – day without discharge
- Dataset used for machine learning modelling:

```
df.head()
```

	date	location	rainfall	discharge	r_d_n-0	r_d_n-1	r_d_n-2	r_d_n-3	r_d_n-4	r_d_n-5	...	d_d_n+1	d_d_n+2	d_d_n+3	d_d_n+4	d_d_n+5	days_since_rain	month	season_str	season_int	target
0	2022-04-01	Newbury	0.29	0.0	0.29	0.29	0.29	0.29	0.29	0.29	...	0.0	0.0	0.0	0.0	0.0	0	4	Spring	1	0
1	2022-04-02	Newbury	0.12	0.0	0.12	0.41	0.41	0.41	0.41	0.41	...	0.0	0.0	0.0	0.0	0.0	0	4	Spring	1	0
2	2022-04-03	Newbury	4.14	0.0	4.14	4.26	4.55	4.55	4.55	4.55	...	0.0	0.0	0.0	0.0	0.0	0	4	Spring	1	0
3	2022-04-04	Newbury	0.21	0.0	0.21	4.35	4.47	4.76	4.76	4.76	...	0.0	0.0	0.0	0.0	0.0	0	4	Spring	1	0
4	2022-04-05	Newbury	0.88	0.0	0.88	1.09	5.23	5.35	5.64	5.64	...	0.0	0.0	0.0	0.0	0.0	0	4	Spring	1	0

Methodology

Exploratory Data Analysis

- Dataset used for machine learning modelling:
  - Data for Newbury.

```
df.describe().round(2)
```

	date	rainfall	discharge	r_d_n-0	r_d_n-1	r_d_n-2	r_d_n-3	r_d_n-4	r_d_n-5	d_d_n+0	d_d_n+1	d_d_n+2	d_d_n+3	d_d_n+4	d_d_n+5	days_since_rain	month	season_int	target
count	1006	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00	1006.00
mean	2023-08-16 12:00:00	2.43	141.74	2.43	4.85	7.27	9.70	12.12	14.55	141.74	283.48	425.21	566.95	708.69	850.43	0.86	6.93	2.45	0.15
min	2022-04-01 00:00:00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00
25%	2022-12-08 06:00:00	0.01	0.00	0.01	0.06	0.29	0.62	1.22	2.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	2.00	0.00
50%	2023-08-16 12:00:00	0.20	0.00	0.20	1.23	2.90	4.93	6.94	9.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.00	2.00	0.00
75%	2024-04-23 18:00:00	2.56	0.00	2.56	6.70	11.35	15.25	19.09	23.13	0.00	0.00	0.00	0.00	0.00	0.00	1.00	10.00	3.00	0.00
max	2024-12-31 00:00:00	38.32	1440.00	38.32	47.73	49.77	63.41	81.87	84.81	1440.00	2880.00	4320.00	5760.00	7200.00	8640.00	14.00	12.00	4.00	1.00
std	NaN	4.79	377.52	4.79	7.48	9.68	11.75	13.59	15.30	377.52	723.49	1052.19	1367.23	1667.90	1956.03	1.98	3.30	1.08	0.36

Methodology

Three different Machine Learning Methods have been used in the analysis.

	Linear regression	kNN - k-Nearest Neighbors	SVM - support vector machines
General	Data scaling is not required (for OLS) but can be done	Data must be scaled and categorical variables converted into numerical.	Data must be scaled and categorical variables converted into numerical.
Model fitting	Model optimisation possible with polynomial features (different degrees)	Tuning with different values for hyperparameter *k*.	Tuning with different hyperparameters *C* and *gamma*.
Pros	• Simple and fast model to implement • Intuitive to interpret • Works well with small datasets	• Easy to implement and fast to train • Intuitive to explain • Suitable for large datasets thanks to non-parametric approach • Applicable for supervised and unsupervised learning	• Powerful and high performing model for variety of tasks • Can work with non-linearities in data • Applicable for regression and classification
Cons	• Assumes linear relationship (in the coefficients) • Sensitive to outliers • More challenging with complex non-linear patterns • Prone to overfitting especially with high degree polynomials	• Difficult to interpret relationships between features and response • Hyperparameter *k* must be optimised • Often not the best performing model • Data preprocessing important (scaling, no categorical features, no missing data)	• Slow training • Parameters are learned but difficult to interpret directly

Results

Comparison of ML models: (Showing model with best results for each type of ML)

Model	Train accuracy	Test accuracy	Precision	Recall
Naïve benchmark	0.8085	0.8085	--	--
Logistic Regression	0.8443	0.8270	0.727	0.268
kNN with k = 12	0.8665	0.8350	0.383	0.684
SVM (kernel='poly', degree=4, C=0.1, gamma='auto')	0.6928	0.8380	0.215	0.372

- All three models have accuracy higher than naïve benchmark; however, differences are very small (max = 0.0295).
- SVM polynomial model shows the highest accuracy but suffers from the lowest precision and recall. This means that predictions quality is poor for precise forecasting of discharges.
- Logistic Regression has relatively high precision which means that predictions can be relied on as there are few false alarms. However, many missed alarms due to low recall.
- kNN model shows high recall what means that predictions are correctly identified as true / false but low precision means that many discharges are not predicted at all.

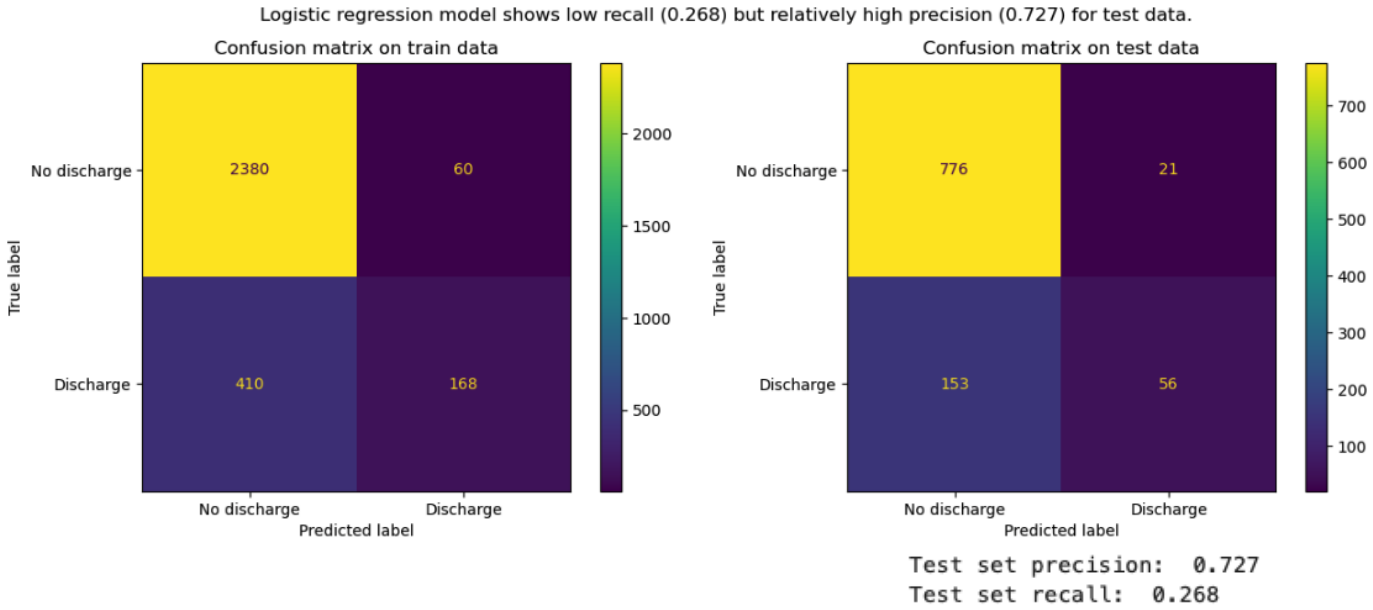
Results

Linear regression ML

- Accuracy scores:

Model	Train	Test
Naïve benchmark	0.8085	0.8085
Logistic Regression	0.8443	0.8270
Quadratic model	0.8416	0.8131
Polynomial degree = 1	0.8443	0.8270
Polynomial degree = 1	0.8416	0.8111
Polynomial degree = 1	0.8472	0.8091
Polynomial degree = 1	0.8403	0.8151
Polynomial degree = 1	0.8373	0.8181
Polynomial degree = 1	0.8396	0.8141

- Confusion matrix for Logistic Regression
(the best performing linear regression model)



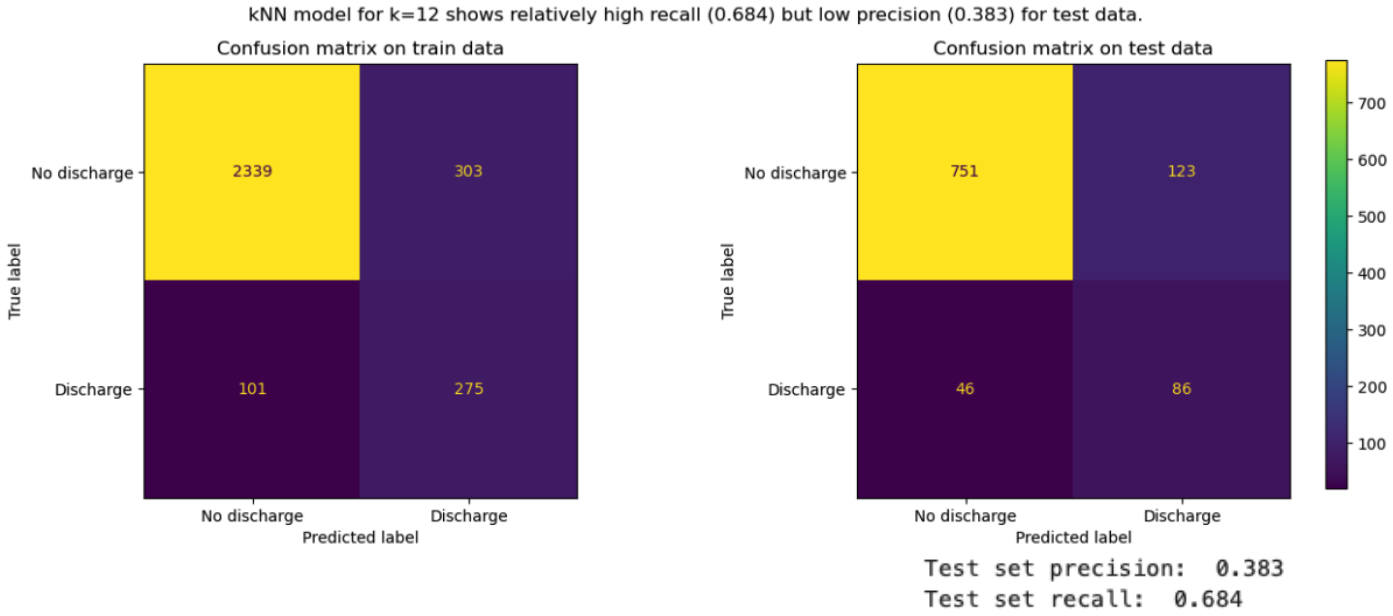
Results

k-Nearest Neighbors

- Accuracy scores:

Model	Train	Test
Naïve benchmark	0.8085	0.8085
k = 1	0.9983	0.7962
k = 2	0.9056	0.8131
k = 3	0.9072	0.8191
…		
k = 11	0.8668	0.8300
k = 12	0.8665	0.8350
k = 13	0.8661	0.8320

- Confusion matrix for k=12
  (the best performing kNN model)



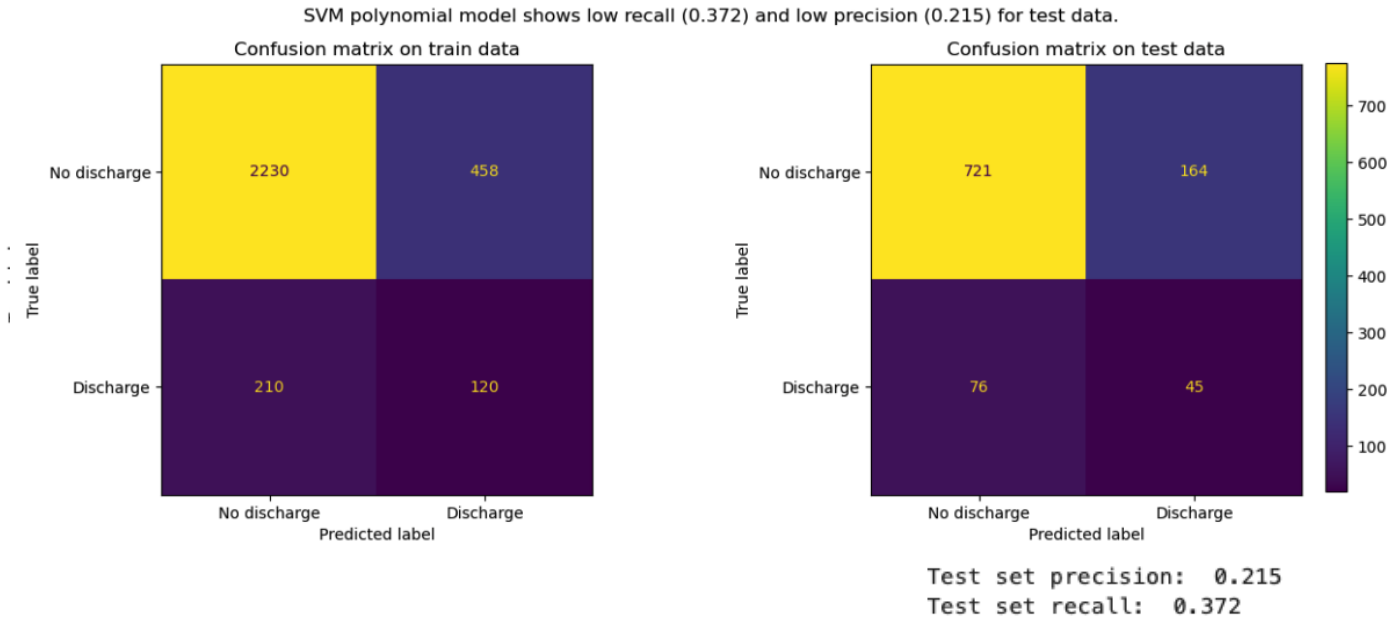
Results

Support Vector Machines

- Accuracy scores:

Model	Train	Test
Naïve benchmark	0.8085	0.8085
SVC	0.8582	0.8300
SVM (kernel='linear', C=1, gamma='auto')	0.8144	0.7873
SVM (kernel='rbf', C=1, gamma='scale')	0.8545	0.8320
SVM (kernel='rbf', C=1, gamma='auto')	0.9450	0.7932
SVM (kernel='sigmoid', C=1, gamma='auto')	0.8085	0.7922
SVM (kernel='sigmoid', C=1, gamma='scale')	0.6928	0.6700
SVM (kernel='poly', degree=4, C=0.1, gamma='auto')	0.6928	0.8380

- Confusion matrix for SVM 'poly' model
  (the best performing model with degree=4, C=0.1)



Conclusions

Analysis of discharges and rainfall shows:

- 0.50 correlation (between total rainfall over the prior 6 days and total discharge duration in the following 2 and 3 days).
- that large discharges (over several days) usually take place in winter.
- Accuracy of reviewed Machine Learning models is better than naïve benchmark but the difference is very small.
- None of the models shows high precision and recall which are required for high quality predictions. Individual models, once optimised, do show high precision or recall but not both.

Recommendations for next steps:

- Explore further Machine Learning models to investigate if quality of predictions improves. Suggested models: decision trees and ensembles.
- Explore if additional variables improve the quality of predictions. Suggested variables: temperature, utilisation of site's capacity.
- Increase sample size to add top sites from all water companies and date range where data is available.
- Add data from all top sites to dataset but consider site's size on likelihood of discharging.