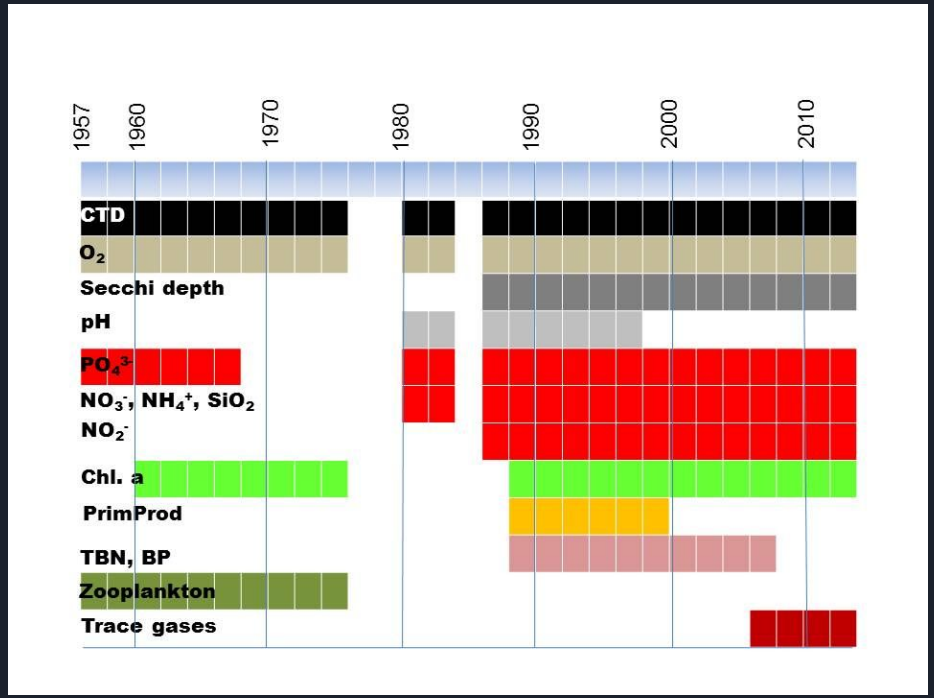
Baltic sea

Wenjie Hu

Background

- Connected with over 200 rivers and Low water exchange with global ocean
- Those elements together and their balance is what matters most for sustainable marine life
- Warming of Baltic Sea water masses since 1850
- Water temperature is an important variable as it greatly affects the functioning of marine ecosystems.



Dataset

Contains measurements of listed parameters collected from baltic sea

- Collected at Boknis Eck observatory since 1957
- Monthly sampling at Boknis Eck began on 30 April 1957.
- Standard sampling depths: 1, 5, 10, 15, 20, 25 m.

element	short-form
Dissolved oxygen	OXY
Phosphate	PO4
Total phosphorus	TP
Nitrogen dioxide	NO2
Nitrate	NO3
Ammonia	NH4
Total nitrogen	TN
Silicon dioxide	SIO2
Chlorophyll	CHLORA
Secchi depth	SECCHI
Dissolved Inorganic Carbon (DIC)	PH
Salinity	CTDSAL
Centigrade Temperature	CTDTMP

	DATE	DEPTH	CHLORA	CTDSAL	CTDTMP	NH4	NO2	NO3	OXY	PO4	SECCHI	SIO2	TN	TP
0	1960-03-31	500	2.9000	17.9	2.2	0.3375	0.0554	0.3171	515.6	0.03	6.235	3.6970	25.1927	0.68
1	1960-03-31	2600	26.5000	20.0	1.5	2.5706	0.1581	1.5025	396.9	0.27	4.636	5.5850	22.1835	2.26
2	1960-03-31	100	2.8000	17.8	2.2	0.5256	0.0826	0.4062	421.9	0.15	6.373	2.7148	24.4312	1.19
3	1960-03-31	2000	7.9000	18.7	1.9	0.4485	0.1080	0.4641	478.1	0.10	4.612	5.6783	26.4591	4.91
4	1960-03-31	1500	6.6000	18.5	2.1	0.5174	0.0873	0.4382	471.9	0.16	5.166	4.5986	24.5783	0.84
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4763	1960-03-02	1500	6.4320	20.5	1.4	2.7990	0.1811	1.4977	375.0	0.55	5.316	8.3581	20.1958	0.84
4764	1960-03-02	2000	12.6118	20.5	1.3	2.8744	0.2598	1.7073	412.5	0.59	4.994	13.9645	22.9771	1.13
4765	1960-03-02	100	2.8478	20.4	1.6	3.4568	0.2934	1.8342	356.3	0.73	6.848	11.9858	20.4109	0.90
4766	1960-03-02	1000	17.7056	20.4	1.5	1.9859	0.1712	1.2574	462.5	0.42	5.363	7.7960	19.9980	0.58
4767	1960-03-02	500	4.2995	20.4	1.6	2.7513	0.1838	1.3876	343.8	0.54	6.410	8.9349	21.7504	1.10

4768 rows × 14 columns

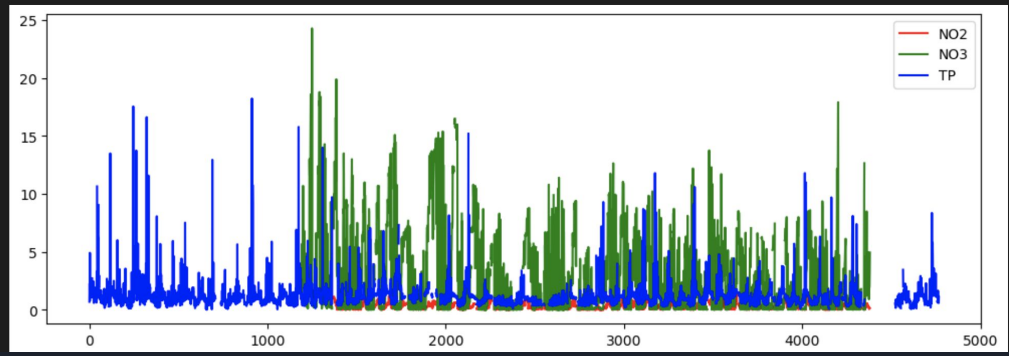
Data example

```
df['OXY']
✓ 0.0s

0       515.6
1       396.9
2       421.9
3       478.1
4       471.9
        ...
4763    375.0
4764    412.5
4765    356.3
4766    462.5
4767    343.8
Name: OXY, Length: 4768, dtype: float64
```

Oxygen

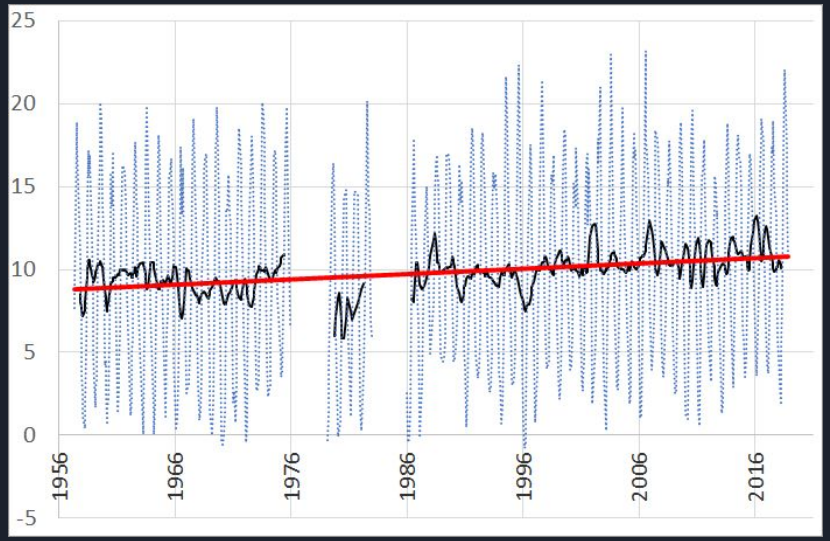
Besides the obvious warming of the surface layer there is a trend to increasing number of anoxia, here defined as O_2 below the detection limit of the Winkler method (i.e. ~2 µM): During the period 1957-1983 only two anoxic events (1979, 1981) have been recorded, whereas during the period 1986-2018 12 anoxic events (1989, 1991, 1992, 1994, 2001, 2002, 2003, 2005, 2007, 2008, 2014, 2016) have been recorded.

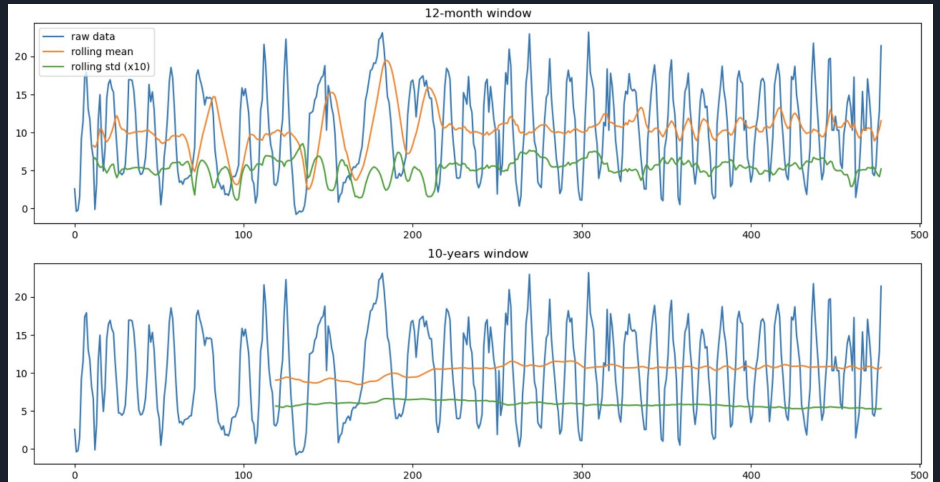
```
plt.figure(figsize=(12, 4))
plt.plot(df['NO2'],'red',label='NO2')
plt.plot(df['NO3'],'green',label = 'NO3')
plt.plot(df['TP'],'blue',label = 'TP')
plt.legend()
plt.show()
✓ 0.2s
Python
```



Temperature

Sea surface temperatures (SST) show a trend to warmer temperatures especially during the last decade which is illustrated by the fact that cold events with winter SST below 0°C have not been recorded since February 1996.

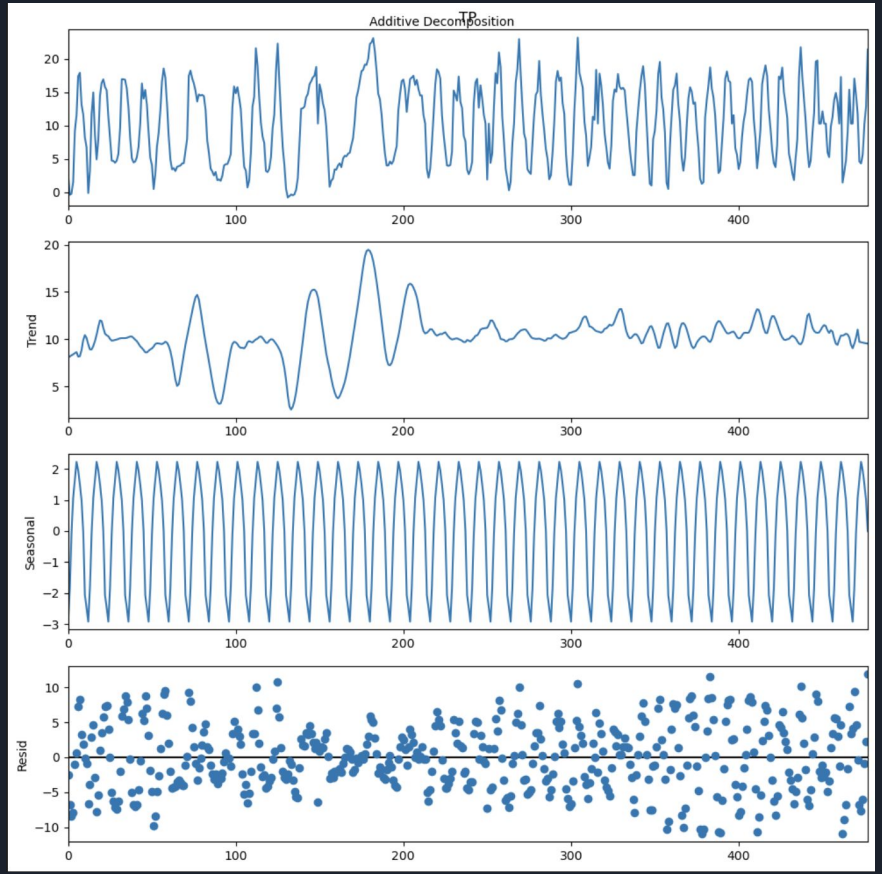




Temperature

We can use Time series decomposition to visualize our data.

- The trend in temperature tends to be stable
- there is a clear seasonality
- The residuals indicate that the series is considered stable



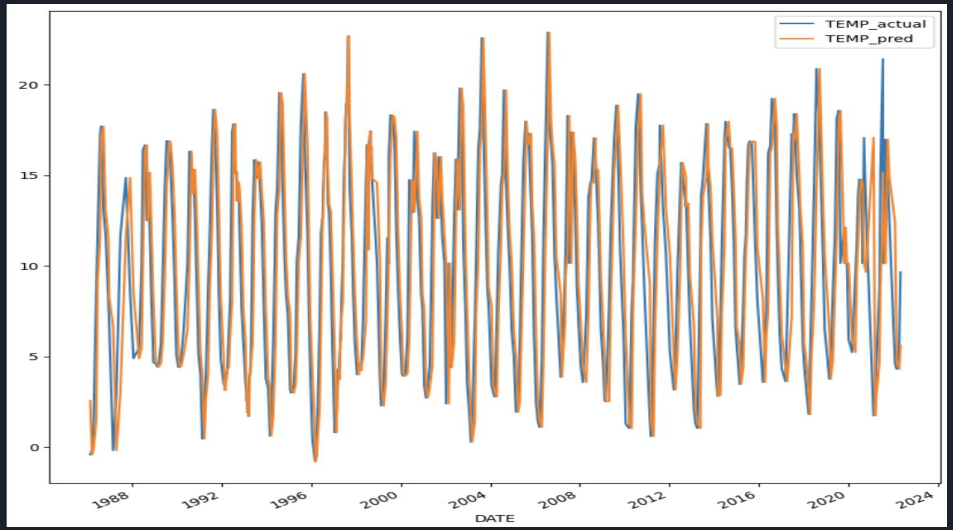
Baseline model

We generate a basic model which uses the current temperature as a forecast for the second month. Thus, we will predict the weather based on the following assumptions: the temperature for this month depends on the temperature of the previous month and so on.
A value of 3.47 for RMSE is acceptable

```
predicted_df = df_.shift(1).rename(columns={'TP':'TEMP_pred'})
actual_df = df_.rename(columns={'TP':'TEMP_actual'})

one_step_df = pd.concat([actual_df,predicted_df],axis=1)
one_step_df.sort_index(axis=0)

one_step_df = one_step_df[1:]
one_step_df = one_step_df.sort_index()

temp_pred_error = MSE(one_step_df.TEMP_actual, one_step_df.TEMP_pred)
print('MSE:', temp_pred_error)
print('RMSE:', sqrt(temp_pred_error))
```

✓ 0.0s

```
MSE: 12.034428804230886
RMSE: 3.469067425725664
```



SARIMA model

use GridSearchCV to find the best P, D, Q value for SARIMA(Seasonal Autoregressive Integrated Moving Average)

- **p**: Trend autoregression order.
- **d**: Trend difference order.
- **q**: Trend moving average order.

```
p = d = q = range(0,2)
pdq = list(itertools.product(p,d,q))
seasonal_pdq = [(x[0], x[1], x[2], 7) for x in list(itertools.product(p,d,q))]

print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
```

✓ 0.0s

```
SARIMA的 examples
SARIMAX: (0, 0, 1) x (0, 0, 1, 7)
SARIMAX: (0, 0, 1) x (0, 1, 0, 7)
SARIMAX: (0, 1, 0) x (0, 1, 1, 7)
```

```
params_list = []
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(one_step_df.TEMP_actual,
                                            order = param,
                                            seasonal_order = param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)

            results = mod.fit(disp = False,full_output=False)
            # print('SARIMA{}x{}7 - AIC:{}'.format(param,param_seasonal,results.aic))
            # case = {'param': param, 'param_seasonal': param_seasonal, 'aic': results.aic}
            # aic_dict.update(case)
            params_list.append([param, param_seasonal, results.aic])
        except:
            continue
```

✓ 6.5s

```
params_list = sorted(params_list, key=lambda x: x[2])[:5]
params_list
```

✓ 0.0s

```
[[(1, 0, 1), (1, 1, 1, 7), 2342.933818458624],
 [(1, 0, 1), (0, 1, 1, 7), 2359.3354725753106],
 [(1, 0, 0), (1, 1, 1, 7), 2384.409647658945],
 [(1, 1, 1), (1, 1, 1, 7), 2399.5601130332143],
```

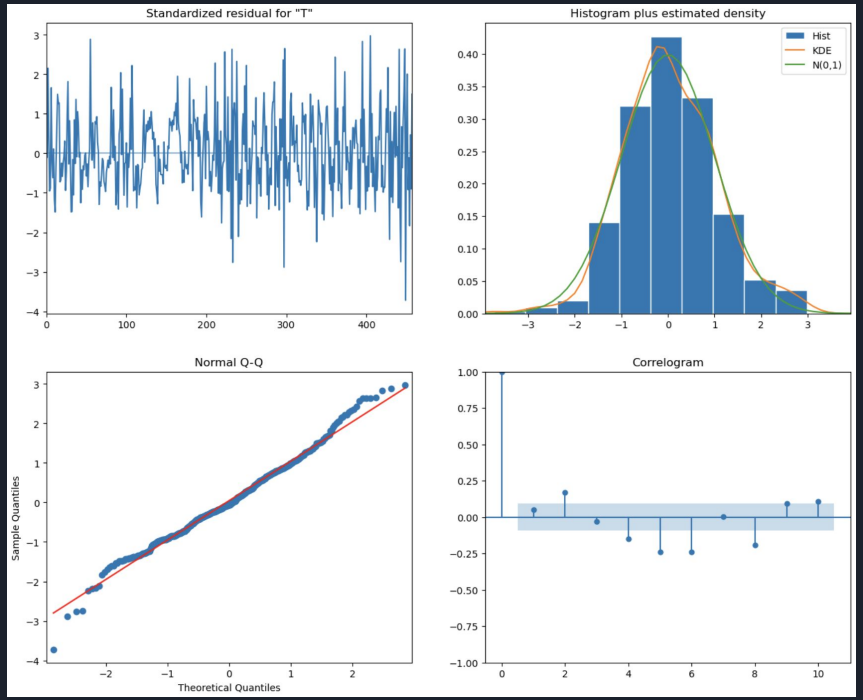
SARIMA model

Use GridSearchCV to find the best P, D, Q value for SARIMA(`Seasonal Autoregressive Integrated Moving Average`)

- **p**: Trend autoregression order.
- **d**: Trend difference order.
- **q**: Trend moving average order.

Fitting the model and diagnose

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.7183	0.043	16.747	0.000	0.634	0.802
ma.L1	0.2987	0.047	6.319	0.000	0.206	0.391
ar.S.L7	-0.2413	0.049	-4.951	0.000	-0.337	-0.146
ma.S.L7	-1.0292	0.035	-29.748	0.000	-1.097	-0.961
sigma2	8.6398	0.750	11.519	0.000	7.170	10.110

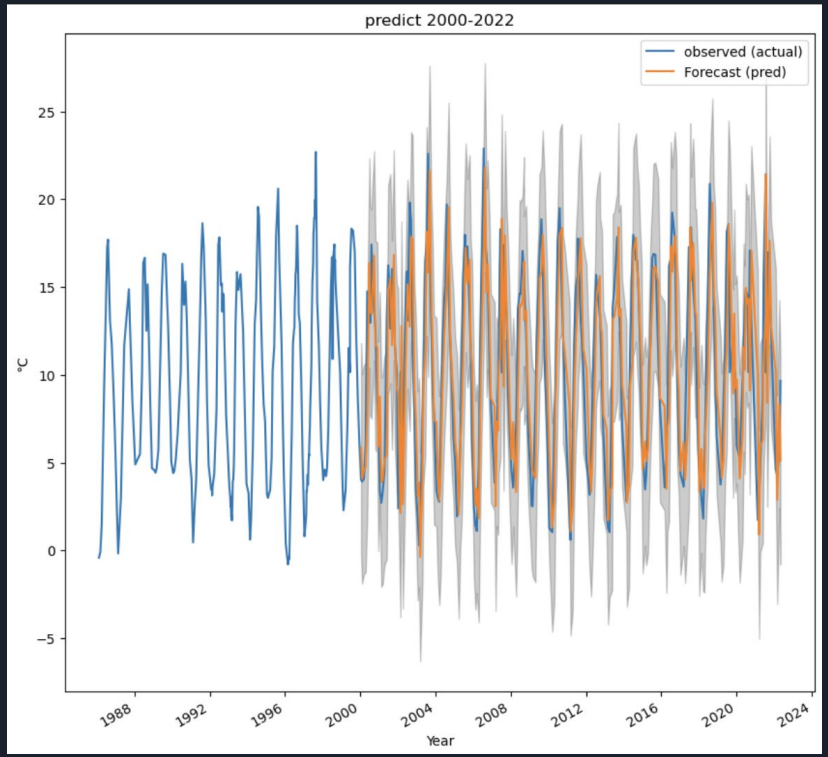


Validating the Forecasts

We used data after 2000 as the test set.

Compare the predicted values with the actual values of the time series, using get_prediction() to obtain the predicted values of the time series and the associated confidence intervals.

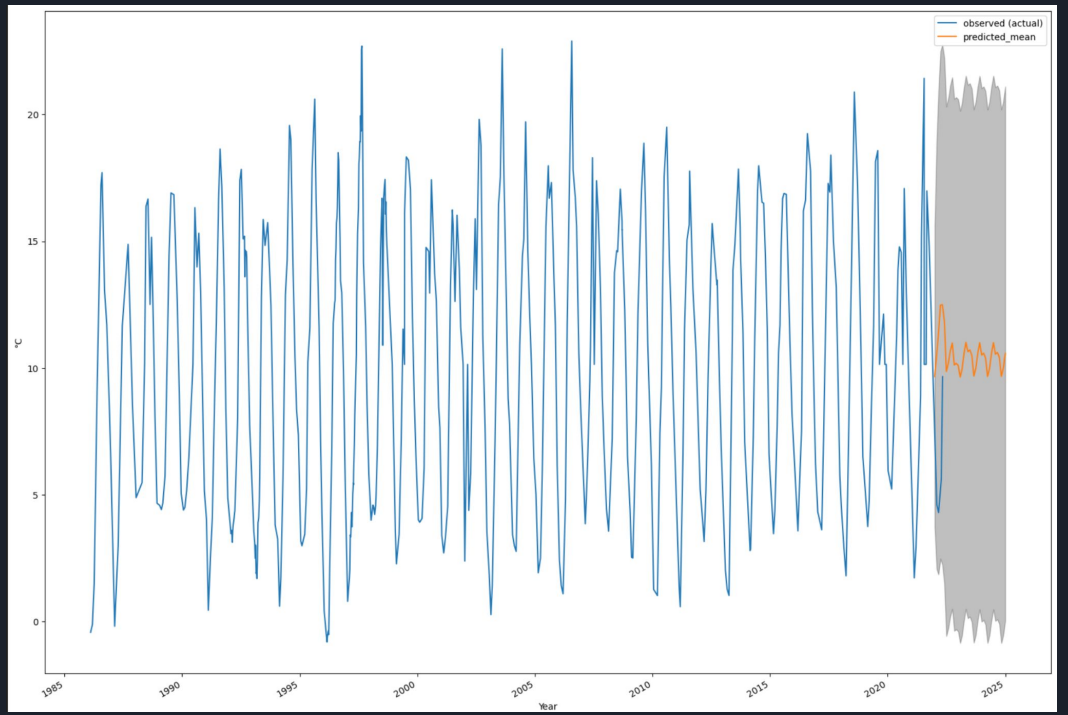
The forecast MSE is 3.4, which is acceptable.



Further forecasts

Using models to predict temperatures over the next 3 years

Also, the model can be easily modified to predict other elements



sources

code:

- https://drive.google.com/file/d/1ICm_z6nhdFfpHo6wbTVWzcsc-98JlOEE/view?usp=sharing

dataset:

- https://www.bokniseck.de/home