# Prediction of sea surface temperature in Helsinki

## 1 Introduction

When doing water sports, like kayaking or wind surfing, it is quite essential to know what the water temperature is depending on the time of the year. The temperature is affected by many factors and predicting it accurately is not straightforward. Here we aim to predict the long term (seasonal) behaviour based on measurements as a function of time and using different types of polynomial regression models. The data set for surface temperature measurements is obtained from Finnish meteorological institute database. In addition to standard polynomial regression, regularization with ridge regression is also applied. Fitted models are compared using different data sets for training, validation and testing and appropriate choice of the model is discussed.

## 2 Problem formulation

The data points are sea surface temperature measurements taken at certain times. The surface temperature observations were downloaded from Finnish meteorological institute service [1] using Suomenlinna's measurement station located in Helsinki. Measurements are available for every half an hour. Measurements during year 2016 were selected and the temperature values measured at noon were taken to represent each day for which the result was available, i.e. during the winter when the sea is frozen there are no measurement available. Total number of data points was 206. There were no missing features or labels in the data set. The dates are transformed into a number indicating the number of day counted from the beginning of year. This number is the feature of the data point and the temperature is the label. This is because date value as such cannot be used as a feature in polynomial regression.

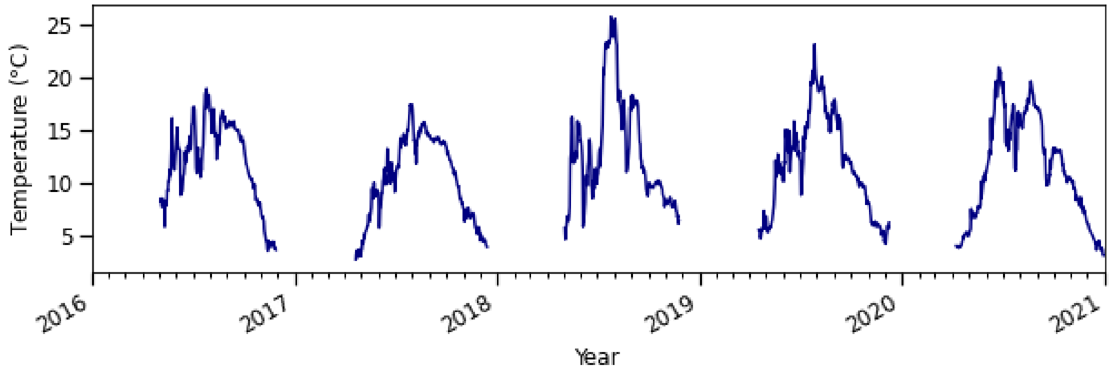

Fig. 1: Surface temperature measurements at Suomenlinna weather station during 2016-2020.

The surface temperatures exhibit quite a complicated behaviour as can be seen from the measure-

👍 Excellent analysis of the ML problem at hand

ments shown in figure 1. The long term seasonal variation between summer and winter is easily observed but additionally there are shorter term variations depending on local climate conditions and these are not directly dependent on the number of day (which is used as a feature). Therefore one cannot expect these simple models to account for short term behavior. The aim here is to try to model the seasonal variation of the surface temperature by using polynomial regression. In a simple model one can imagine the surface temperature to follow a parabolic curve attaining its maximum during the summer and decreasing towards autumn and fall (until it reaches zero degrees). It can be also seen that linear regression is not an appropriate model and would be unphysical as well. In addition to 2nd-order parabolic curve higher order polynomials are also explored.

👍 Very logical explanation on why polynomial models are good fits

## 3 Methods

🤔 What is the motivation behind using a single split?

Sci-kit learn package [2] was used for fitting the models, Pandas package [3] was used for preprocessing of the data and Matplotlib package [4] for producing plots. Data points are divided randomly into training (75%) and validation sets (25%) and these sets used throughout to fit different models. Different orders of polynomials ($d = 2 \ldots 10$) are fitted to the training set by minimizing the mean squared loss [5]. The mean squared loss is calculated by summing over all data points $\{x_i, y_i\}_{i=1}^{N}$ and obtaining the squared difference for the predicted value given particular hypothesis (i.e. polynomial) $h(x) = \sum_{j=0}^{D} c_j x^j$

$$L_{\mathrm{MSE}}(h) = \frac{1}{N} \sum_{i=1}^{N} \big(y_i - h(x_i)\big)^2 \tag{1}$$

The same loss is also calculated for both the training and validation sets with different fits. Comparison of loss values gives an indication how well the model fits the data and overfitting can be suspected when the difference between these loss values starts to increase.

As higher order polynomials may result in overfitting, regularization is also considered in order to alleviate the overfitting problem. In practice this is done by imposing an additional constraint to the mean squared loss consisting of L2-norm of the polynomial coefficients scaled by a numerical parameter [5]. This procedure is known as ridge regression. The loss for ridge regression can be written as

$$L_{\mathrm{Ridge}}(h) = \sum_{i=1}^{N} \big(y_i - h(x_i)\big)^2 + \alpha \sum_{j=1}^{D} c_j^2 \tag{2}$$

👍 Well-explained motivation to include regularisation for high degree polynomials. The discussion shows good understanding on the potential weakness of the model.

where $c_j$ are the polynomial coefficients of the hypothesis $h$ and $\alpha$ is the regularization strength (hyperparameter). The constant coefficient $c_0$ is not regularized. Note also that here the ridge loss is not calculated as a mean anymore, i.e. no division by $N$. The scaling parameter was set to its default value ($\alpha = 1$); other settings (e.g. 5,0.2) were also tried but they either did not change the results or gave unreasonable results, i.e. constrained the coefficients too much.

## 4 Results

The models obtained for different polynomial orders are shown in figure 2 including training and validation data. The mean squared loss calculated for training and validation data sets with different models is also shown in figure 2.

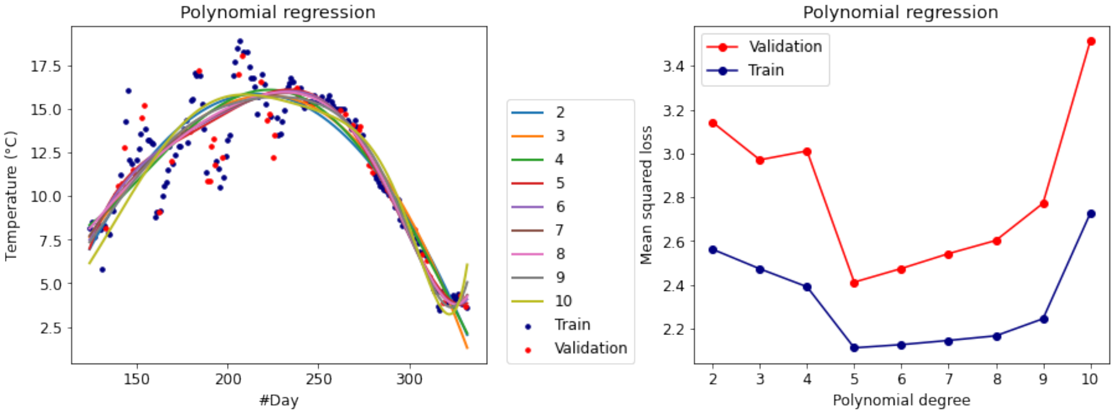

Fig. 2: Polynomial regression fit for different orders and loss calculated with training and validation sets.

One can observe that 5th-order polynomial has lowest loss and for higher order polynomials the loss starts to increase. For validation set this is expected behaviour due to the overfitting but for training set one would expect the loss either to decrease or at least to stay the same, i.e. even if higher order terms are added, setting the new coefficients to zero should give the same results as with the lower order polynomials. However, the loss seems to increase slightly even for the training set. This could due to some numerical issues as well. One way to constrain the polynomial coefficients is to use regularization, i.e. in addition the mean squared loss one simultaneously minimizes L2-norm of the polynomial coefficients. In this way one can influence the model so that even if the polynomial degree increases, the solution remains at least of similar quality as with the lower order polynomials.

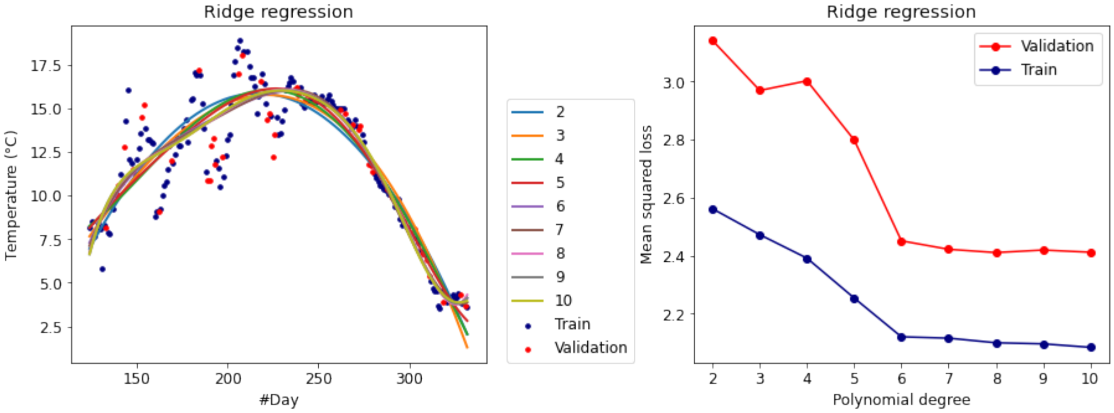

Fig. 3: Ridge regression for different orders and loss calculated with training and validation sets.

The same analysis using the ridge regression is given in figure 3. Although the results do not differ much, one can observe that after 6th-order both loss values basically stay the same. This can be understood by thinking the new higher order coefficients being close to zero and the fitted model basically staying the same even if the order of the polynomial is increased. However, regularization does not offer any significant improvement of the results and therefore is not used for the final model. Based on these results, 5th-order polynomial is chosen as the final model. In order to test the model with unseen data and as the aim has been the prediction of long term behavior of surface temperatures during the year, the corresponding measurements from year 2017 are used as test data (there are 238 data points in total). These results are shown in figure 4. The mean squared error for the final model and the test data is 3.52.

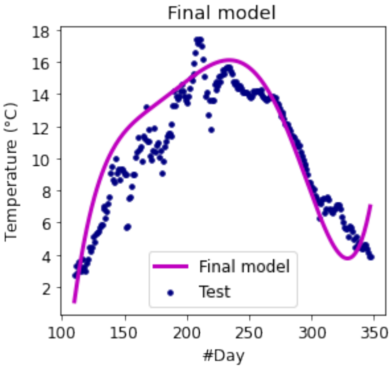

Fig. 4: Final model compared to the test data.

One can observe that the model fits relatively well the test data, showing the slower temperature increase during the spring and more rapid temperature decrease during the autumn quite accurately. Of course, there are differencies in the absolute values and also unphysical temperature increase in winter time. This is probably explained by the fact that the training set extended only until the end of November whereas in the testing set the measurements continued until mid December. Higher variations during the spring could be explained by the fact that occasional storms mix warmer surface water with colder water but during autumn water has similar temperature throughout and similar effect is not so clearly present.

One can also study the accuracy of an old folk lore that water temperatures start to decrease 25th of July (on Jaakko's day). The maximum of the model is attained on 234. day which is 23. of August which is roughly a month later and therefore the model does not support the traditional belief. Of course, this shift could be due to the climate change as well and could be worth a further study.

## 5 Conclusions

Polynomial regression is a very simple approach and it is clearly not able to account all the variability in the data. For example, including seasonal variation (summer/winter) by considering data from several years simultaneously is not very straightforward with polynomial regression. Additionally, there is also the problem of missing data points during the winter time, i.e. to replace those with zeros (ice covered surface) is not necessarily the best solution. One could think of some kind of periodic function (i.e. like Fourier-series) for the model to account for several years.

Making predictions within the data, e.g. interpolation, is possible although not very reliable due to the short term variations. One method which could improve the predictability is using Gaussian processes [6] with an appropriate kernel, e.g. a spectral mixture kernel [7] which could account for correlations between different length scales in the data and enable the extrapolation, i.e. to make predictions outside the data as well. In this kind of approach one could include data from several years. This surface temperature data is actually an example of a time series data [8] and there are number of different types of models which can be used for analysis and prediction.

## References

[1] https://www.ilmatieteenlaitos.fi/havaintojen-lataus

[2] https://scikit-learn.org/stable/index.html

[3] https://pandas.pydata.org/pandas-docs/stable/index.html

[4] https://matplotlib.org/index.html

[5] S. Raschka and V, Mirjalili, *Python Machine Learning - Second Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*, Packt Publishing (2017).

[6] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, The MIT Press (2005).

[7] A. Wilson and R. Adams, *Gaussian Process Kernels for Pattern Discovery and Extrapolation*, Proceedings of the 30th International Conference on Machine Learning, **28**, 1067-1075 (2013).

[8] https://en.wikipedia.org/wiki/Time_series