Data Analysis Project 2

MA8701 Advanced Methods in Statistical Inference and Learning V2021

Florian Beiser, Yaolin Ge & Helene Minge Olsen

03 May, 2021

In this project, we analyse a real dataset using methods from part 2-4 of the MA8701 course.

Note on Open Science

To pursue the idea of reproducible research, the chosen dataset as well as the code for our analysis are publicly accessible:

- dataset: https://github.com/metno/havvarsel-data-driven-pred
- code: https://github.com/FlorianBeiserNTNU/MA8701

The Data Set

For this project, we generated a data set ourselves by fetching real-world weather observations from the `Havvarsel-Forst` and `Frost` database servers of the Meteorological Institute of Norway (MET). To this end, we established a routine downloading from the respective APIs, which can be found in the aforementioned repository. Since the data was obviously not analyzed before in a similar way, the subsequent result will contribute to active research.

The constructed data set consists of multiple time series with measurement of different weather properties and a time series for the water temperature at *Sjøstrand* in Asker kommune during the summer months, which was originally collected through "badevann.no" from 2016 til 2020. Sjøstrand is a popular swimming site south of Oslo, hence, the study of its water temperature is interesting for the general public and recreation industry.

As index, our data set contains the times of measurement which are typically made at 4 full hours per day. Technically, the water temperatures are collected in the respective column

- `water_temp` in `degC` measured at 7, 10, 14, 16 every day during swimming season.

The goal of the data analysis is to predict the swimming tempertures at Sjøstrand beach using atmospheric weather values (no forecasted values) from surrounding weather observation stations. In the aforementioned data generation process via the `Frost` database of MET we retrieve values for 7 different atmospheric weather values where we identify the closest stations for those measurements respectively and include them in our dataset. Thereby, we follow the naming convention: "station_id" (identifyer of the station where the measurement is done) + "element" (atmoshperic quantity that got measured) + "number" (0=measurement in 2m height, 1=measurement in 10 height above ground). The lonlat-coordinates of the relevant stations can be found in the `log1.txt` file in the repositories. As elements we consider:

- `air_temperature` in $degC$
- `wind_speed` in m/s
- `cloud_area_fraction` in $0, ..., 8$ (0 no clouds, 8 fully cloudy)
- `mean(solar irradiance)` in W/m^2 - a quantity for the intensity of the sun
- `sum(duration of sunshine)` in min - sunshine in the last hour

- `mean(realtive humidity)` in
- `mean(downwelling shortwave flux)` in W/m^2 - a quantity for the intensity of the sun (similar to irradiance but only for highly intesive UV radiation)

The data generation process can be repeated (or modified) by running the `run_example.sh` in the data generation repository - however, the dataset which we subsequently use is also provided stationary in the code repository.

Therewith, we end up a data set with a time index, one `water_temp` measurement, and 14 atmospheric measurements.

Model Building

Our goal is to predict the water temperature at the next time using all information from the current time - this includes the `water_temp` at that current time. Therefore, we introduce all atmospheric observations *and* the water temperature at time t as covariates in order to model the response, which is the water temperature at time $t + 1$. (Note that we explicitly assume here that we do not use forecasted values for the atmospheric variables even if that would be a valid data source in further research projects. Moreover, we ignore that the time difference between the measurements is bigger during the night than during the day to keep the model simpler.)

In a nutshell, the dataset consists of 1 response, 15 covariates and 2327 observations.

Preprocessing

Even if atmospheric measurements may exists for more times than the water temperature, we restrict ourselves to those time points when water temperature measurements exists. However, it happens that the measurement equipment is out of order for a short period of time (it happens but it happens only very seldomly and then typically one day of atmospheric observations at a station is missing). This data is *missing completely at random* and within the data generation process it is imputed from its nearest neighbor. Even though this method is commonly not recommended we argue that the weather does not completely change within a few hours and this guess is better than mean imputation (maybe in Norway the weather changes quicker than in other parts of the world, but only very very few atmospheric measurements are missing such that we do not expect any influence on the data analysis.)

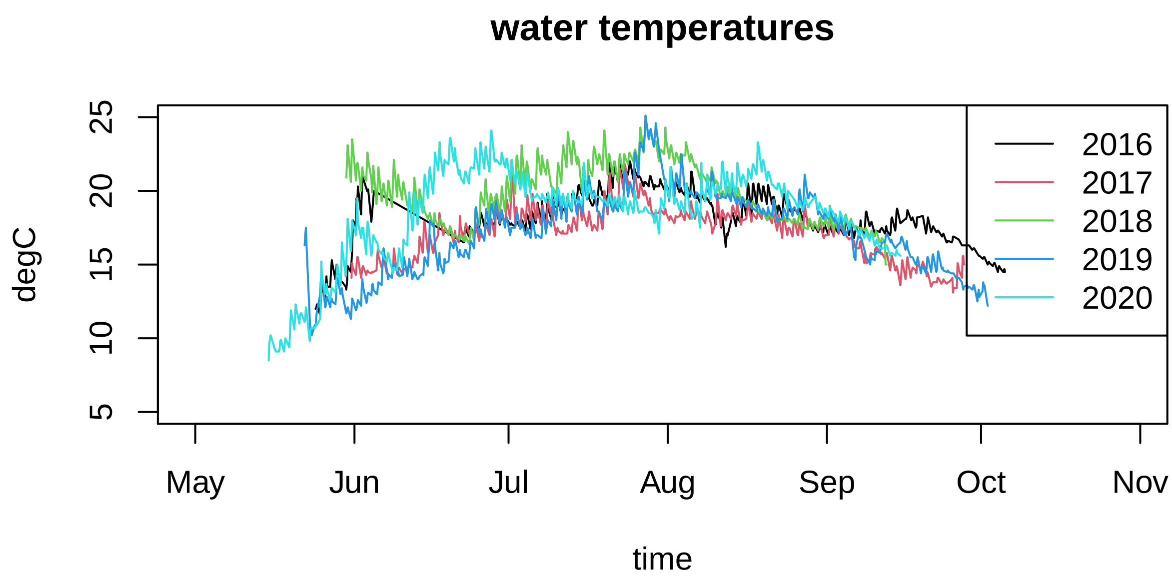
The data covers 5 years such that we use the years 2016-2019 (about 80% of the data) as training set in the data analysis and we set the last year (2020, corresponding to about 20% of the data) apart as test data to validate the results.

```
##                        time water_temp SN19710air_temperature0
## 2 2016-05-24 10:00:00+00:00       12.0                     9.8
## 3 2016-05-24 14:00:00+00:00       12.1                    13.0
## 4 2016-05-24 16:00:00+00:00       12.3                    12.4
## 5 2016-05-25 07:00:00+00:00       11.6                    11.9
## 6 2016-05-25 11:00:00+00:00       12.6                    15.4
## 7 2016-05-25 13:00:00+00:00       12.9                    17.5
##   SN19923air_temperature0 SN19430air_temperature0 SN19940air_temperature0
## 2                     9.0                    11.7                    11.1
## 3                    12.7                    14.5                    14.2
## 4                    12.5                    14.5                    13.9
## 5                    10.0                    13.1                    11.6
## 6                    15.0                    17.2                    16.9
## 7                    17.1                    19.3                    18.9
##   SN19923wind_speed0 SN19940wind_speed0 SN18700cloud_area_fraction0
## 2                3.6                1.4                           8
## 3                4.2                1.4                           7
## 4                5.0                1.5                           7
```

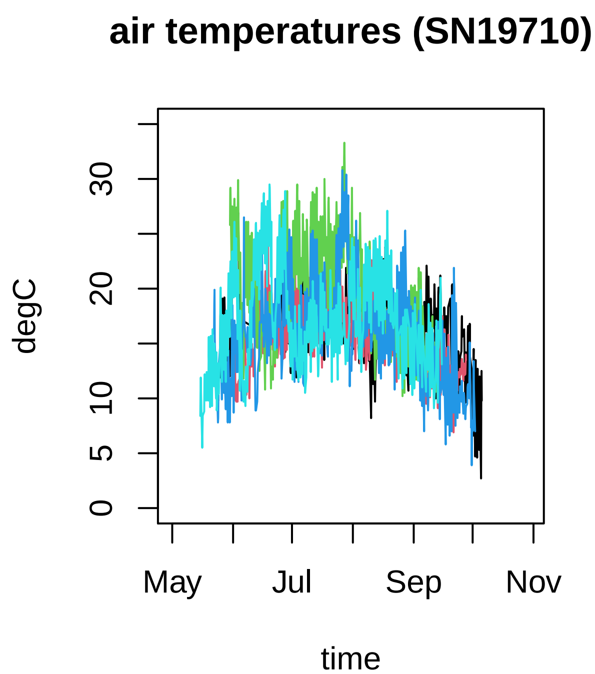
```
## 5              3.4               1.8                         1
## 6              3.9               1.4                         1
## 7              3.7               1.2                         1
##   SN17150cloud_area_fraction0 SN28380cloud_area_fraction0
## 2                           8                           5
## 3                           8                           5
## 4                           8                           3
## 5                           8                           4
## 6                           8                           1
## 7                           8                           1
##   SN50539mean.solar_irradiance.PT1H.0 SN11500sum.duration_of_sunshine.PT1H.0
## 2                               0.737                                      0
## 3                               0.737                                      0
## 4                               0.737                                      1
## 5                               0.737                                      0
## 6                               0.737                                     60
## 7                               0.737                                     60
##   SN19940mean.relative_humidity.PT1H.0 SN26950mean.relative_humidity.PT1H.0
## 2                                   83                                   76
## 3                                   66                                   78
## 4                                   64                                   69
## 5                                   59                                   55
## 6                                   48                                   48
## 7                                   42                                   43
##   SN19940mean.surface_downwelling_shortwave_flux_in_air.PT1H.0 response
## 2                                                        200.9     11.7
## 3                                                        330.0     12.0
## 4                                                        281.6     12.1
## 5                                                        392.8     12.3
## 6                                                        801.0     11.6
## 7                                                        763.4     12.6
```

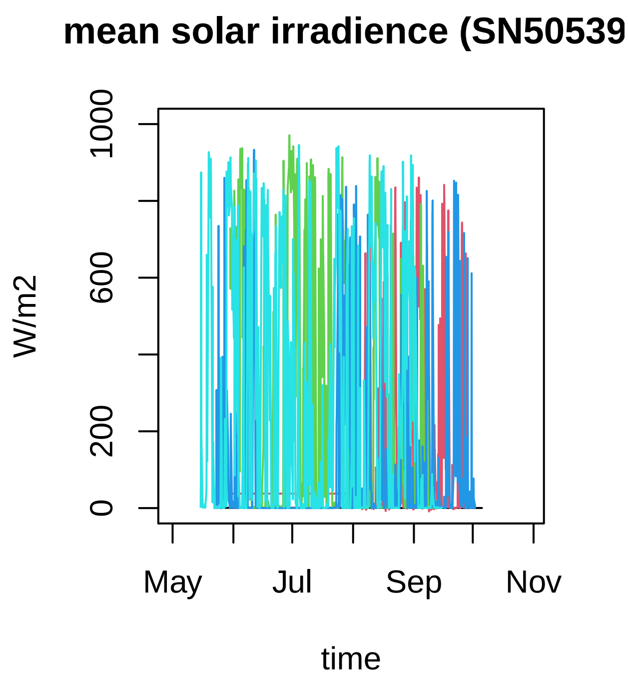
Data Exploration

We start the exploration by investigating the response.

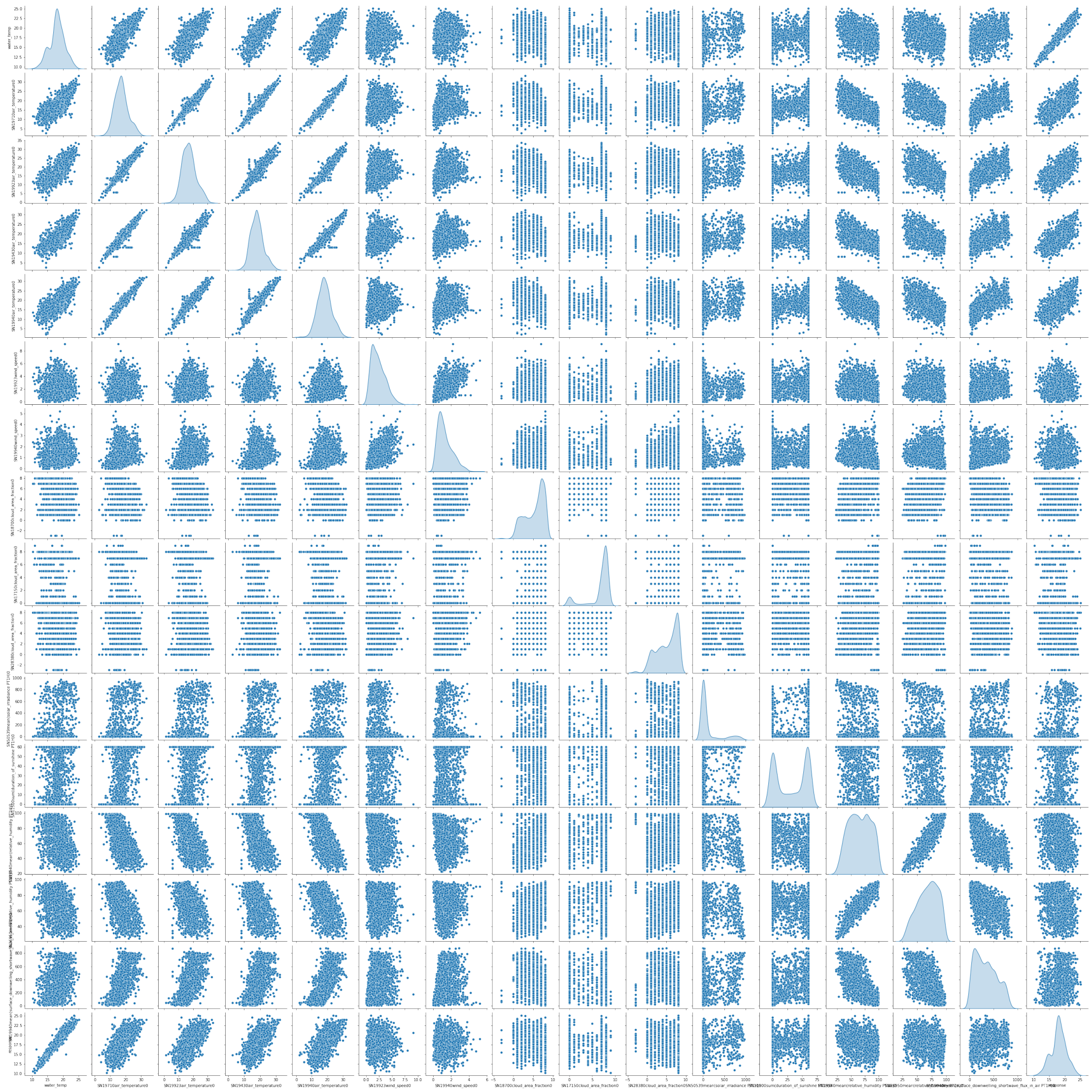


The water temperature shows a very pronounced seasonality in its progression. Moreover, it changes only rather slow from one time to the next.





In some of the covariates this seasonality is also present, but with an much higher volatility, and in some covariates we cannot recognize any significant seasonality at all and additionally they change much faster - as example we show the air temperature and the solar irradiance averaged over the last hour at their observation station which is closest to Sjøstrand.



Further the seaborn pairplot (see python part of this project) shows sometimes clear linear correlation between the covariates and the response, but also between the covariates themselves, which is “dangerous” as we have seen in Kjerstis lectures. However, also very unclear relations between the response and the covariates are visible.

This yields that non-linear data analysis methods are needed for the dataset at hand.

Random Forest Model Fit

Since the water temperature changes rather slowly compared to the atmospheric quantities and we have the water temperature of the previous time included as covariate, we know that this strong predictor `water_temp` will dominate the trees produced if we choose to use bagging. To obtain decorrelated trees and improve the variance reduction, we therefore want to fit a random forest model to our data set.

We utilize the `randomForest` package in R. As the random forest allows a random selection of m covariates $m \leq p$ to be considered for the split for each node, a 5-fold cross validation was applied to find the optimum number of covariates. As the data set contains 19 variables, the general rule for regression is to set the number of randomly chosen variables to be considered as $floor(\frac{19}{3}) = 6$. However, we obtain the smallest cross-validation error for $m = 4$. This can be explained by our data set being very correlated, and it is therefore wise to set m to be small.

```
## [1] 16  8  4  2  1
```

```
##
## Call:
##  randomForest(formula = response ~ . - time, data = data, mtry = 4,      importance = TRUE, subset =
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 0.4632432
##                     % Var explained: 92.47
```

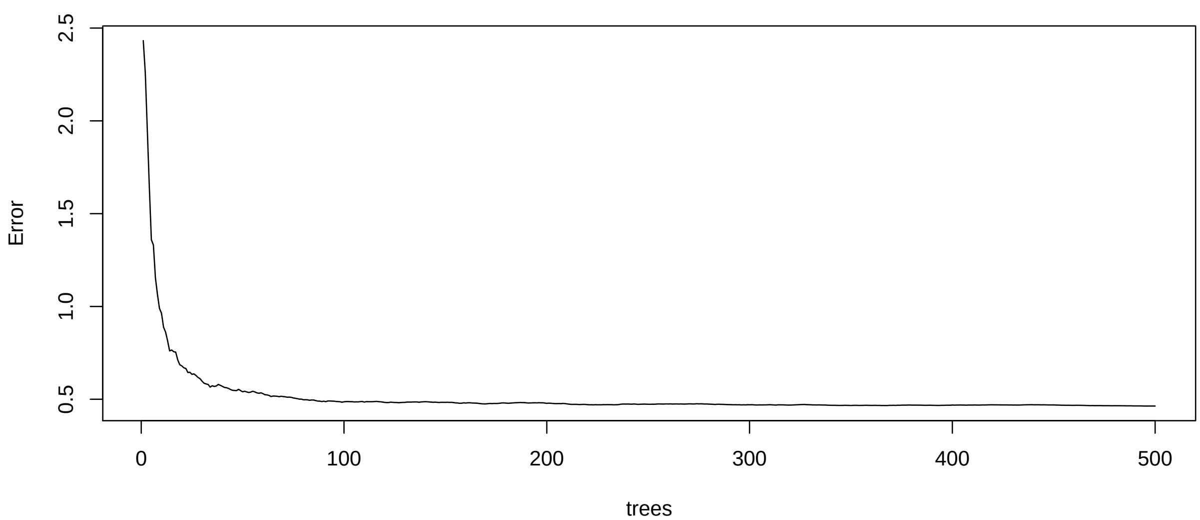


Figure 1: Plot of numbers of trees

As each tree uses a different bootstrap sample, the OOB sample is used as a validation set. From Figure 1 we observe how the OOB error estimate is reduced as a function of the nr of trees. As a result, the model fits 500 trees to average over, with a percentage of variance explained, also known as pseudo R-squared, around 91.97%

Figure 2 depicts a good fit except for a slight overprediction for lower temperatures between 9 and 13 degrees. This may corresponds to the fact that in this low-temperature region less observations are available.

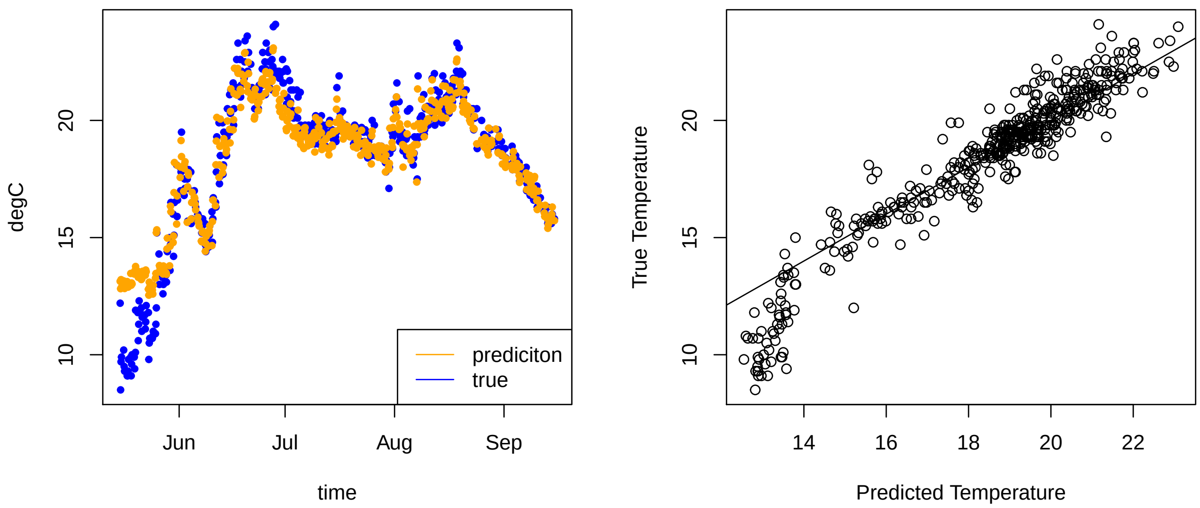


Figure 2: Predicted test values against true values.

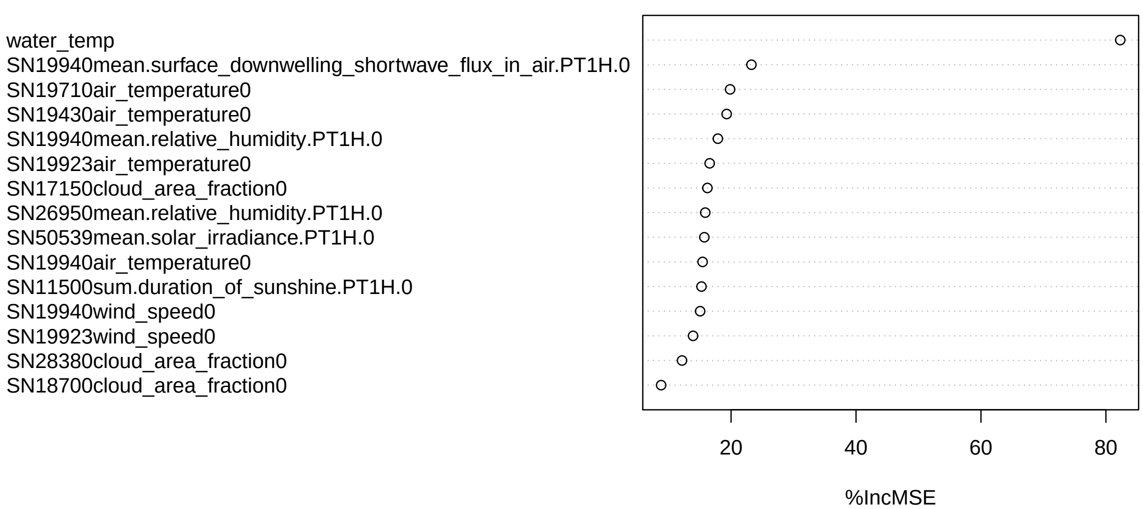


Figure 3: Variable importance plot based on increase in MSE when premuted and tested using OOB sample, higher values indicate larger impact when premuted, hence larger importance.

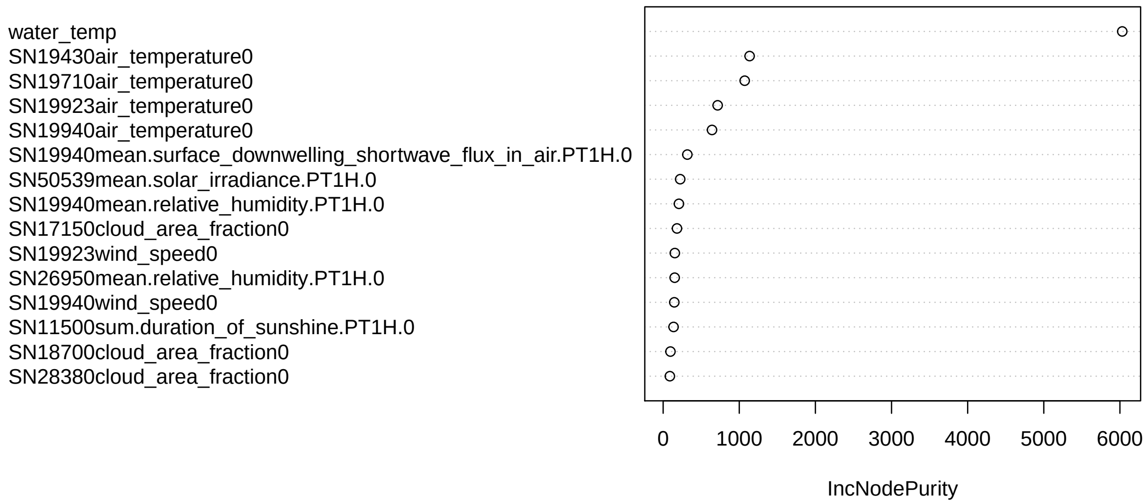


Figure 4: Variable importance plot based on increase in node purity, higher values indicate larger importance.

Figure 3 and 4 depicts that the variable importance plot based on the OOB sample tends to spread the importances more uniformly, but `water_temp` remains a main predictor for both. We observe that there are a lot of predictors considered to be relevant after `water_temp`, which indicates a well performance from the random forest model. This is reflected in the low test MSE.

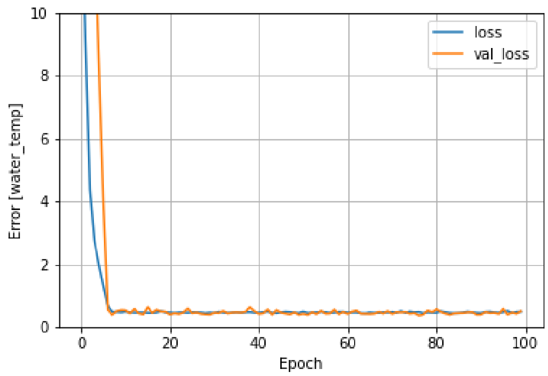
```
## Test MSE of Random Forest Model
##  1.243495
```

Neural Nets

The code can be found in `Project3.ipynb`.

Due to complexity restrictions, only one neural network is going to be trained and tested here.

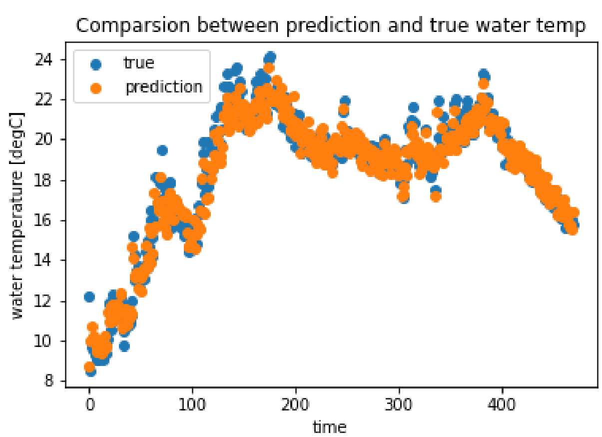
Training

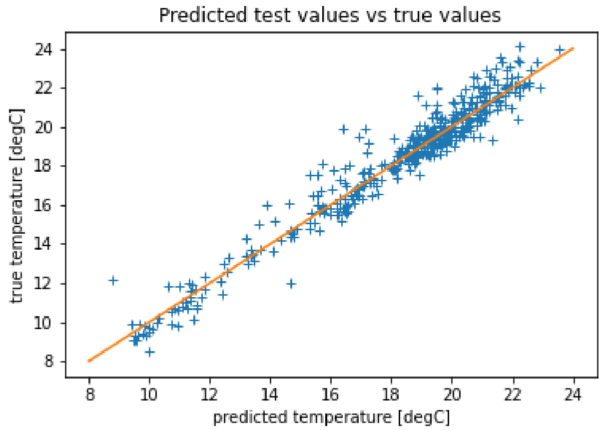


The training history plot yields that the training is pretty fast to converge - the loss is minimized quickly. This is one the of the benefit for using the simple model. `Loss` represents the loss during the training while `val_loss` represents the loss for the validation set during the training.

Prediction

We evaluate the performance of the trained neural net on the test set which are the observations from the last year.





The MSE for the prediction sets is 0.61352, meaning that the prediction deviates in average by half a degree, what will be barely feelable for the swimmer.

The trained neural net shows a good performance on the test set. The prediction seems with low variance and no clear bias. In particular, the neural nets extrapolates better for low temperature prediction, which only arise in the test but not in the training set.

Conclusion

We have fetched a dataset from different sources, which contains the water temperature at a popular swimming site in Asker and collected atmospheric measurements from the MET weather observation stations around. The goal of our project work was to do prediction for the water temperature of the next future time using all information of the current time. The data exploration suggests the need for non-linear data analysis method, wherefore we have fit a random forest model and a neural net. Both approaches show a grandious prediction quality. The random forest is a bit biased for low temperatures, but in contrast to the neural net it allows us to identify the most significant variables, which coincide qualitatively to a large extend with our intuition.