Assignment III
Intelligent Systems

Forecasting

This assignment will be discussed on January 27, 2020

As usual, we offer a practice session next Monday 18, 2:15 pm instead of the regular lecture. Please, use the new Zoom link for this:
https://uni-kiel.zoom.us/j/86200263127?pwd=aE4vTGllWGlSbWRxV0lqc3pFTVRPZz09

Overview

This is the last of three assignments. It is again mandatory for each group to give a brief presentation and hand in your results – in a slightly different form (see below).

FT I	Preprocessing	✓	Mo, 23.11.2020
FT I	Preprocessing Presentation	✓	We, 25.11.2020
FT II	Feature Selection	✓	Mo, 14.12.2020
FT II	Feature Selection Presentation	✓	We, 16.12.2020
FT III	Model Selection	✓	We, 13.01.2021
FT III	Model Selection Presentation		We, 27.01.2021

Goal

For this last assignment we want you create a model based on the provided data from the stations to predict the water level of the main station. You have free choice which method to use. On the one hand this involves "free" research on your part, but on the other hand you are allowed to use existing libraries. Finally, we want you again to give a short presentation and provided a program that predicts water levels for unknown data.

Data Basis

Again, we use the already known data – with one important alteration: As it became clear in the last assignment that station B (*Oberstorf* in Bavaria) has little or no relevance it has to be left out. Therefore, only use the data for the main station (*Willenscharen*) and the station A (*Padenstedt*) and station C (*Itzehoe*) To start, you have two options:

A. Rerun your own steps for preprocessing from the first assignment with the reduced input data.

B. Use the preprocessed dataset `AS_3_preprocessed_water_time_series_data.tbz` provided by us.

Either is then the basis for further steps, like feature selection or segmentation.

Model Creation

For the actual forecasting, you can use concepts from the lecture and also go beyond scope of the lecture (so far) and research your own approach for creating a model. We propose to look at a standard library like *Scikit Learn*:

https://scikit-learn.org/stable/user_guide.html

It offers tools for predictive data analysis. Starting points here would be "Supervised Learning" on regression and "Model Selection".

Training and Testing

You can use the provided tools from Sckit Learn to select and train a model. Remember to divided your input data into a training and test set. You can try out and choose the distribution and other model parameters. There are two requirements:

A. Train your model to predict for a sample of 7 days the water level of the main station of the following day at times 3 hours apart: 00:00, 03:00, 06:00, . . .

B. As loss function for training use *Root Mean Squared Error (RMSE)*.

See also *Evaluation* below.

Presentation

Again, we want you make a give a small presentation, which should address the following points:

- What steps (preprocessing, feature selection, segmentation, model selection, . . .) did you take?
- And why?
- How did you choose your training and test data set? And what stop criterion?
- How does your model perform on the input data?

Evaluation

You might have noticed that the youngest provided data samples are from *2017-12-31*. We want you hand in a program that – based on your model above – forecasts future water levels of the main station. For this we have (held back) data, ranging from *2018-01-01* to *2019-12-31*. This target data has the same format and resolution as the input data.

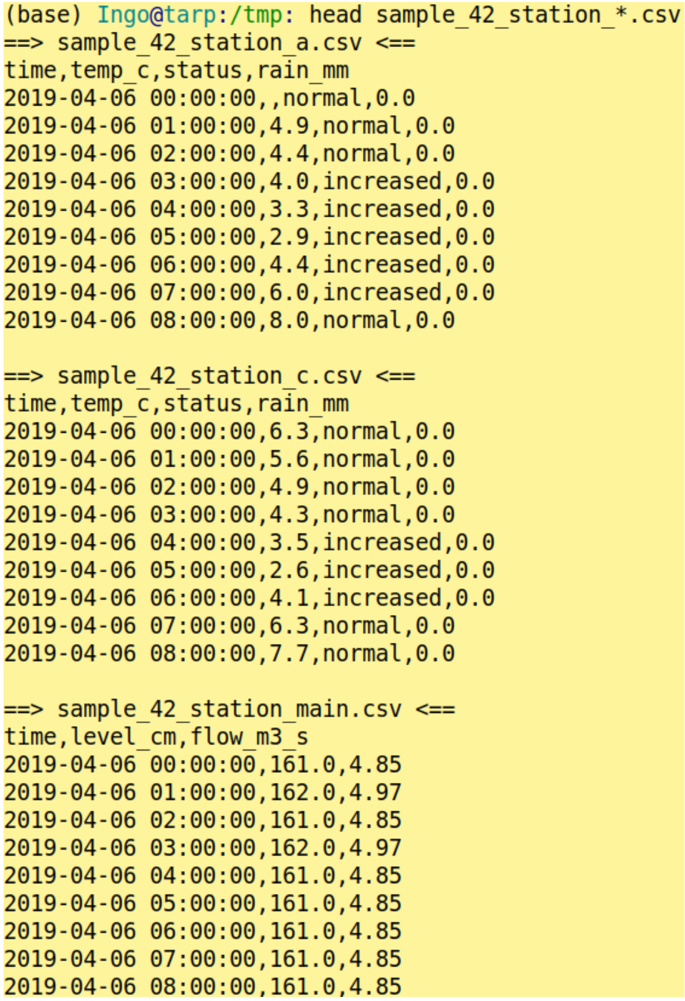
Predictor

Please provide (via OLAT) a program for prediction with the following requirements:

- The predictor can be a Jupyter notebook or a standalone Python program.
- Your predictor has to run without any parameters.
- It takes all input samples in the current directory to for prediction
- As resulting prediction it creates on output file in the current directory.

Input Samples

Each input sample X contains data for 7 full days (e. g. *2019-04-06 – 2019-04-12*) and consists of 3 files: `sample_X_station_a.csv`, `sample_X_station_c.csv` and `sample_X_station_main.csv`. For example:



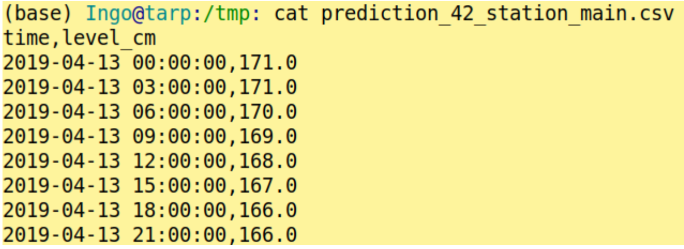
```
(base) Ingo@tarp:/tmp: head sample_42_station_*.csv
==> sample_42_station_a.csv <==
time,temp_c,status,rain_mm
2019-04-06 00:00:00,,normal,0.0
2019-04-06 01:00:00,4.9,normal,0.0
2019-04-06 02:00:00,4.4,normal,0.0
2019-04-06 03:00:00,4.0,increased,0.0
2019-04-06 04:00:00,3.3,increased,0.0
2019-04-06 05:00:00,2.9,increased,0.0
2019-04-06 06:00:00,4.4,increased,0.0
2019-04-06 07:00:00,6.0,increased,0.0
2019-04-06 08:00:00,8.0,normal,0.0

==> sample_42_station_c.csv <==
time,temp_c,status,rain_mm
2019-04-06 00:00:00,6.3,normal,0.0
2019-04-06 01:00:00,5.6,normal,0.0
2019-04-06 02:00:00,4.9,normal,0.0
2019-04-06 03:00:00,4.3,normal,0.0
2019-04-06 04:00:00,3.5,increased,0.0
2019-04-06 05:00:00,2.6,increased,0.0
2019-04-06 06:00:00,4.1,increased,0.0
2019-04-06 07:00:00,6.3,normal,0.0
2019-04-06 08:00:00,7.7,normal,0.0

==> sample_42_station_main.csv <==
time,level_cm,flow_m3_s
2019-04-06 00:00:00,161.0,4.85
2019-04-06 01:00:00,162.0,4.97
2019-04-06 02:00:00,161.0,4.85
2019-04-06 03:00:00,162.0,4.97
2019-04-06 04:00:00,161.0,4.85
2019-04-06 05:00:00,161.0,4.85
2019-04-06 06:00:00,161.0,4.85
2019-04-06 07:00:00,161.0,4.85
2019-04-06 08:00:00,161.0,4.85
```

Prediction Output

The output file has to be named `prediction_X_output.csv` and contain the 8 water level predictions (e. g. for *2019-04-13*):



```
(base) Ingo@tarp:/tmp: cat prediction_42_station_main.csv
time,level_cm
2019-04-13 00:00:00,171.0
2019-04-13 03:00:00,171.0
2019-04-13 06:00:00,170.0
2019-04-13 09:00:00,169.0
2019-04-13 12:00:00,168.0
2019-04-13 15:00:00,167.0
2019-04-13 18:00:00,166.0
2019-04-13 21:00:00,166.0
```

These example files are available in OLAT.