EMPOLIS

Data Science Challenge

Hydropower is one of the most attractive forms of sustainable energy. Hydropower plants account for over 70% of all renewable electricity in the world today. Such plants are normally operated in different operation modes (e.g., depending on the power requirements over a day or week). Understanding transitions between these operation modes is of high importance for machine diagnosis.

In the attached file “*hydropower_time_series.7z*”, you’ll find a data set that shows the output power generated by a plant over a period of one month. Alongside the output power, there is also information about the measured sound pressure in the power plant at different frequencies over the same period. A more detailed description of the data set is provided in the table below. Please note the sampling rate of the different fields.

Data Description

Field	Definition	Unit	Sampling Rate
`timestamp`	Sampling timestamp	datetime (s)	-
`power`	Power generated by the plant	MW	1 Hz
`octave_62_5`	Measured sound pressure at 62.5 Hz	Pa	0.1 Hz
`octave_125_0`	Measured sound pressure at 125 Hz	Pa	0.1 Hz
`octave_250_0`	Measured sound pressure at 250 Hz	Pa	0.1 Hz
`octave_500_0`	Measured sound pressure at 500 Hz	Pa	0.1 Hz
`octave_1000_0`	Measured sound pressure at 1000 Hz	Pa	0.1 Hz

Task

Devise a model that detects transition conditions in a hydropower plant based exclusively on the available sound pressure values. Effectively, given a time series of sound pressure levels at the above-mentioned frequencies, your model should output whether the plant is in transition or not at each timestamp.

A plant is said to have transitioned from one mode to another if the power value increases/decreases by at least +/- 1 MW over a window of at least 10 seconds and then stabilizes. Longer transition periods are possible if the value continues to increase/decrease. Note that fluctuations in the power value should not be considered as transitions: the new power value should be at least +/- 1 MW compared to the power value before transition.

You may use any approach and any appropriate metric to develop and test your detection model. Your model should be able to analyze a new test data set and output the transition state (e.g., 0 and 1) for each timestamp. It is important to clearly show all steps that you perform in your analysis; the classification accuracy is secondary. Note that there are missing/wrong data in the provided data set.

Functional Programming Challenge

Industrial machines generally follow certain processes based on values stored in one or more pre-defined configuration sheets. Deviations from these settings can indicate/lead to failure or wear and tear of the machine components. It is paramount to observe any changes in these processes and identify potentially dangerous developments in advance.

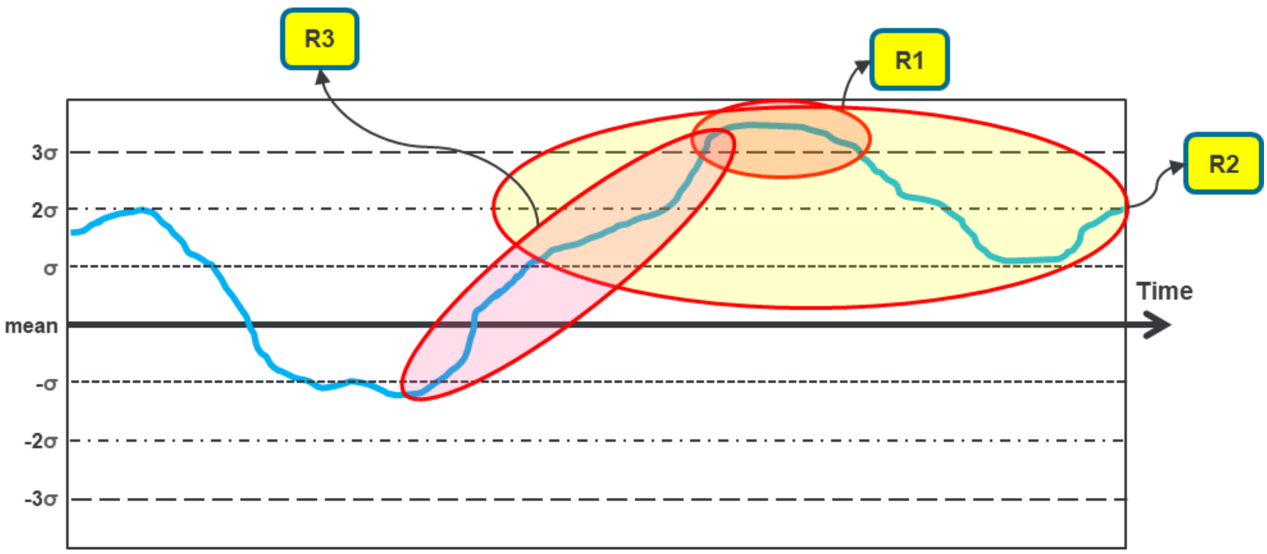
Nelson rules (proposed by Lloyd Nelson) are a method to observe a process and determine whether some measured variable is exhibiting abnormal characteristics, or in other words, out of control. These are a set of 8 rules that are based on the mean and standard deviation of the measured variable. The rules are applied to a ***control chart*** that shows the magnitude of a measured variable against time.

The 8 rules are as follows:

1) *Out of 3 sigma*: One point is more than 3 standard deviations from the mean.
2) *Bias*: Nine (or more) points in a row are on the same side of the mean.
3) *Trend*: Six (or more) points in a row are continually increasing (or decreasing)
4) *Alternating*: Fourteen (or more) points in a row alternate in direction, increasing then decreasing
5) *Out of 2 sigma*: Two (or three) out of three points in a row are more than 2 standard deviations from the mean in the same direction.
6) *Out of 1 sigma*: Four (or five) out of five points in a row are more than 1 standard deviation from the mean in the same direction.
7) *Low deviation*: Fifteen points in a row are all within 1 standard deviation of the mean on either side of the mean.
8) *High deviation*: Eight points in a row exist, but none within 1 standard deviation of the mean, and the points are in both directions from the mean.

Example Illustration

In the below illustration, the blue curve is a time series. The y-axis is centered around the mean of the time series, while the dashed lines in either direction denote multiples of its standard deviation. Each point on the blue curve is relative to the computed mean. R1, R2 and R3 are Nelson rules 1, 2, and 3 from the list above.



Task

In the attached file "*nelson_rules_time_series.csv*", you will find a time series with the first column being the timestamp and the second column the corresponding sensor value.

Your task is to design a nelson rules monitor that detects and identifies the first 4 Nelson rules – "*out of 3 sigma*", "*bias*", "*trend*", and "*alternating*" – and outputs the active Nelson rules at each timestamp. Note that if a point is part of a Nelson rule (e.g., one of 6 points in a trend [rule 3]), then that Nelson rule is active at the corresponding timestamp.

Your output should look like this:

```
timestamp,rules
…
100,1;2;3
101,1;2;3
102,2;3
…
```

Please note the following criteria:

1) The focus of this exercise is functional programming:
    a. Use pure functional constructs and pre-existing library methods as far as possible
    b. Do not mutate your variables once declared (all declarations should be final)
    c. Do not use any mutable constructs
    d. If you really must use mutable variables, keep them local to a function
    e. Your functions should not produce any side effects
2) Compute the mean and standard deviation based on the first 100 points and use them as the basis for the control charts

Deliverables

For both the tasks, you may choose your programming language and tools as you please. Please send us the following deliverables as part of your submission:

1) Solution file for the second task
2) Your code
3) A small description of your code
4) Instructions on how to set up and run your code
5) Also give us an insight into your methodology and your ideas to help us understand how you came to your solutions, what challenges did you face, what limitations they have, and what possible extensions they could have.

We are looking forward to your submission!

Your Empolis Team