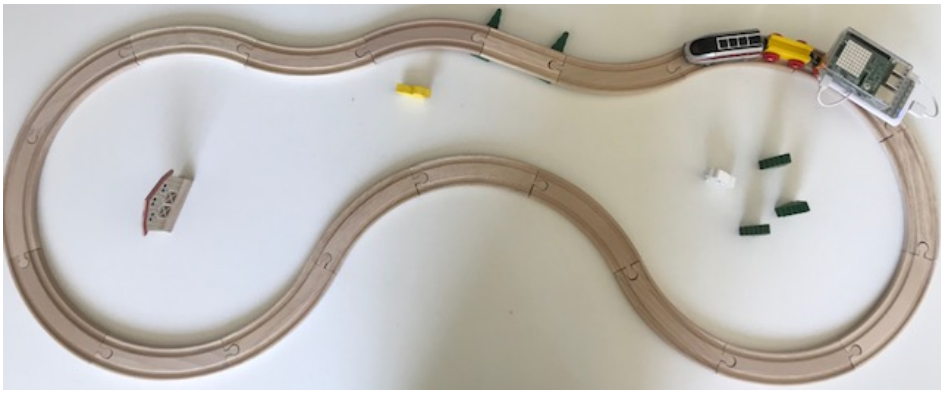
Application assignment for

Machine Learning Engineer

The purpose of the following assignment is to give you an opportunity to show us your skills within machine learning. There is not a single correct answer but you should be prepared to motivate your design choices. When you are ready, you should put your answers on the question below in an email and send it to jobs@ekkono.ai. When we have reviewed your assignment will come back to you. Good luck!

Use case - Toy train PdM

In this assignment you will work with a dataset collected from a toy train riding along a track. It was collected using a Raspberry Pi, a Sense HAT (an add-on board with sensors) and a battery powered toy train.



The train rides on the track seen in the pictures below and it pulls the Raspberry Pi, mounted on top of a power bank, in one of the wagons. On the track there's a small bridge (located close to the yellow duck) that introduces a small uphill and downhill.

Data is collected during eight laps where the following sensor values are collected twice per second.

- **temperature** - The temperature on board the Raspberry Pi
- **pressure** - The atmospheric pressure
- **humidity** - Humidity as measured by the sense hat
- **gyro pitch** - The pitch (up / down) angle of the Pi. Zero degrees indicates a horizontal Pi.
- **gyro roll** - The roll in degrees where zero indicates a horizontal Pi.
- **gyro yaw** - The yaw in degrees from 0 - 360, similar to a compass reading.

At first glance this might seem like a very trivial example, but in fact it has many properties that are present in most real world scenarios. First, each lap can be seen as one process cycle with varying operational settings. Secondly, the process cycle is quite stable delivering similar values each round. Thirdly, the sensors are not always well calibrated.

Objectives

The problem you should solve is to **predict the gyro yaw ten steps into the future**. You are free to use any ML-library and programming language that you like but you are not allowed to use a workbench e.g. Knime since programming will be a daily activity for you. **Don't use Long short-term memory (LSTM) neural networks** to solve the problem.

You should hand in the following:

1. The best ML model you can create for the problem.
2. Your best estimation (using an appropriate error metric) of your models performance on future data.
3. The script containing the code you used to train and evaluating your data.
4. A chart showing the importance of the difference attributes used in the model.