Objective:
During this activity you will learn how to process data to reveal useful information. Remember the sensors you installed previously? We are going to use those sensors, and the values they gathered to predict the amount of rain in the next day. Some important considerations before we begin.

First, there is not enough time to explain all the details regarding machine learning (ML) and predictions, but almost all libraries related with ML are self contained. This means you can provide parameters (such as the average temperature and pressure for a day) and the respective output value (such as the amount of precipitation) and the libraries will be able to try to learn a pattern and reproduce it. Second, as you will see for the overall performance of the system, the most important part is not the learning algorithm, but the initial processing of the data.

Background:
In this activity we are going to use regression algorithms to predict the amount of rain for the following days. We gathered data from 18 of August up until now, and we will try to predict a weather phenomenon (in our case, precipitation). We gathered data related with: temperature, humidity, atmospheric pressure and the amount of rain.

Since the data is continuous, this type of regression is known as time series prediction. This simply means we are going to use the values from instant t, t-1, t-2, …. to learn a phenomenon. Another point to keep in mind is that when the weather changes, more than one parameter changes with it (atmospheric pressure and temperature can drop and that indicates rain). We have prepared the data to allow the learner algorithm to see these changes.

To accomplish these we will use a method known as rolling the dataset. We gathered a dataset for a single period (say, 8 hours), then each line is merged with the previous to create a dataset with period T (the original 8 hours), period T-1 (the previous 8 hours), up to period T-n (further into the past). This way the learner as the ability to see changes from a period to another.

More info in:
https://medium.com/making-sense-of-data/time-series-next-value-prediction-using-regression-over-a-rolling-window-228f0acae363
https://machinelearningmastery.com/time-series-forecasting-supervised-learning/

Activity:
We prepared all the code related with ML on the “tasks.py” script, and given a dataset (the data you pre-process) it tries to predict the amount of rain in the following days. The code uses 4 different learners and shows the mean square error (MSE) of each one. An ideal MSE is close to 0.

The code should also print a graph with the expected value for an entry (in bold red) and the predicted values from each model (the low amount of days used for the first task may return an empty graph).

It is important to notice that the amount of data is relatively small, and it was captured mostly during summer (each means small to zero amount of rain in portugal). Without more data a reliable rain predictor can not be built.

For more information on ML and the learners:
http://scikit-learn.org/stable/tutorial/basic/tutorial.html

Tasks:

1. Using mongodb queries create a summary for a single day (min, max, average...) of each phenomenon (temperature, humidity and pressure). Create a dataset, each row should contain a summary the corresponding value of precipitation. See the results of your prediction.
2. Combine several summaries by adding more days to the dataset. See the predictors performance.
3. Alter the function roll_dataset to create a time series dataset with several periods. Use it (start with only roll=2) and evaluate your predictions.
    a. **Hint:** When you use a roll value of 2, you do not process the last 2 periods in the dataset because you don't have enough data to add to them. Then, from the current day you're processing, you add to it the values from the previous N=roll dataset entries. (Note that the most recent entry is at position 0).
4. Combine several summaries and periods, and see how the performance of the learners evolve.
    a. **Hint:** Try dividing the day into periods starting at 6h, 14h and 22h (8 hours each) to simulate the morning, afternoon and night.
5. Optimize the algorithm (data pre-processing only) to achieve the highest possible accuracy (MSE close to 0).
    a. **Hint:** A day can be partitioned in any number of periods.