Machine Learning for IoT
Lab 2 – Pre-processing

Exercise 1: Timeseries pre-processing

1 On the RPI, continuously run the T/H monitoring script developed in *LAB 1 – Exercise 2*.
2 Set the following retention rules on your Redis TimeSeries database:

a. Set the retention period of the *mac_address:*temperature and *mac_address:*humidity timeseries to 1 day.

b. Create two timeseries with a chunk size of 128 bytes, called *mac_address:*temperature _avg and *mac_address:*humidity _avg, that store the average over every 30s of *mac_address:*temperature and *mac_address:*humidity , respectively. Set the retention period of the two timeseries to 30 days.

c. Create a timeseries with a chunk size of 128 bytes, called *mac_address:*temperature _min, that stores the minimum over every 1 minute of *mac_address:*temperature . Set the retention period of the timeseries to 30 days.

d. Create a timeseries with a chunk size of 128 bytes, called *mac_address:*temperature _max that stores the maximum over every 1 minute of *mac_address:*temperature . Set the retention period of the timeseries to 30 days.

e. For all the timeseries, check the number of samples and their memory size after at least 15 minutes of monitoring.

f. Repeat the steps a., b., c., and e., disabling compression (create new timeseries for each step adding "_uncompressed" to the original names). Report the results in the following table and discuss the collected statistics:

Timeseries Name	**# of Samples**	**Compressed Size (KB)**	**Uncompres. Size (KB)**
*mac_address:*temperature			
*mac_address:*humidity			
*mac_address:*temperature _avg			
*mac_address:*humidity _avg			
*mac_address:*temperature _min			
*mac_address:*temperature _max			

g. For each timeseries, estimate the maximum number of samples and the maximum memory size, considering both the compressed and the uncompressed versions. Report the results in a table.

h. Plot the created timeseries in Deepnote.

<u>Exercise 2:</u> Profiling Audio Features Extraction on the edge

1. On the RPI, create and run a script to measure the execution time (i.e., the latency) needed to process audio feature extraction methods.

a) Download to the RPI the *preprocessing.py* script from Deepnote. Store the script in your working directory.
b) Use the script *lab2_ex2.py* (available in the Portale della Didattica, under the folder *Labs/lab2-preprocessing*) to measure the latency for computing the spectrogram.
c) Modify the script to measure the latency for processing the log-Mel Spectrogram and MFCC.
d) For all the three feature extraction methods, try different hyperparameters values in the following ranges
   - STFT frame length ∈ [10, 50] ms. Try also with power-of-two values like 8ms, 16ms, 32ms, and comment the results.
   - STFT frame step such that overlap ∈ {0%, 25%, 50%, 75%}.
   - # of mel bins ∈ [10, 128].
   - Mel lower frequency ∈ [0, 80] Hz.
   - Mel upper frequency ∈ [2000, 8000] Hz.
   - # of MFCCs ∈ [10, # of mel bins].
e) Report the collected values in a table, following this template:
f) Discuss the collected results.

Method	**Hyperparams values**	**Latency (ms)**
Spectrogram	frame_length=10, frame_step=50%	
Spectrogram	frame_length=10, frame_step=75%	
MelSpectrogram	…	
MFCC	…	
…	…	