Case Study - Machine Learning Engineer

Dreem

Context: We want to create, train and serve a sleep stage classification model for the Dreem headband. The case study will take you through some steps that you may meet as an ML engineer at Dreem. Note that the code quality will also be part of the assessment. You can share the result of your case study with us through a Github/Gitlab repository. After getting the case study, you will have seven days to send it back to us.

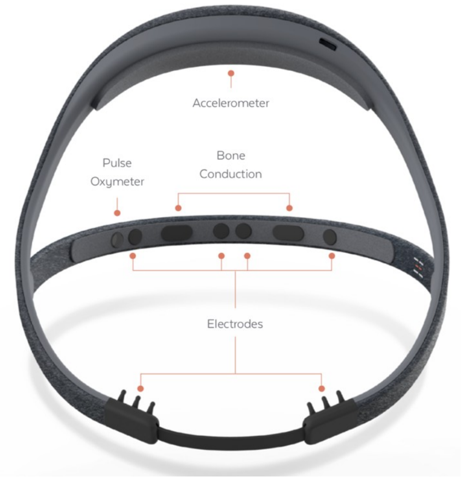

Figure 1: The Dreem headband

Data description

In order to train the sleep staging model, nights are performed both with polysomnography (PSG) and with the Dreem headband. The PSG is scored by sleep experts according to the AASM rules, five stages are defined: Wake, N1, N2, N3 (Deep Sleep), and REM (paradoxical sleep). Each sleep stage is defined by its [frequency contents and specific patterns]() and characterizes 30 seconds of signals. The frequency content of EEG during sleep is typically contained between 0.5Hz and 35Hz, the amplitude of a filtered EEG signal is often between -300 uV and 300uV.

The output of the scoring process is called a hypnogram and defines sleep stages over the complete night.

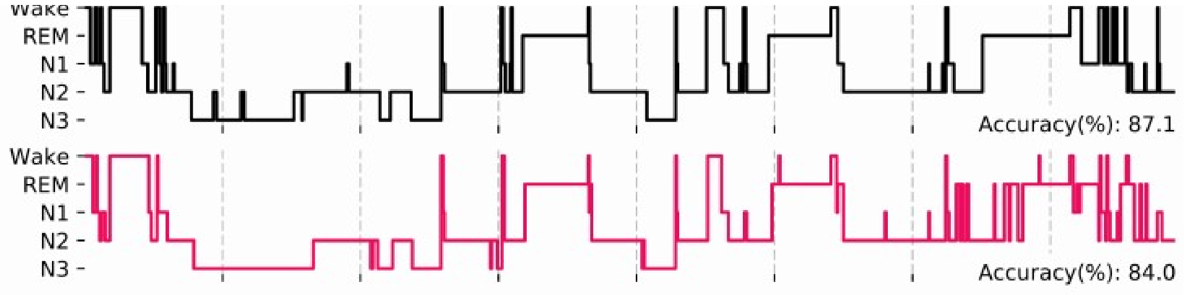

Fig 2: Hypnogram of a night scored by a human expert (Top) and by the Dreem Headband (Bottom)

Part 1 - Data exploration and model training

We provide seven nights from the Dreem headband with their associated hypnogram. Each night is defined by a unique id, the raw data are stored under h5 format and the hypnogram in .json format. The h5 are stored in the h5 folder and the hypnograms in the hypnograms folder.

The h5 file has a dict like structure and contains the following keys:

- eeg1, eeg2, eeg3, eeg4, eeg5, eeg6, eeg7: EEG signals from the Dreem headband sampled at 250Hz.
- accelerometer_x, acclerometer_y, accelerometer_z: accelerometer data from the Dreem headband sampled at 50Hz.

The json file is a list of sleep stages, each sleep stage lasts 30 seconds, so the first sleep stage corresponds to EEG data from 0 to 30 seconds, the second stage to EEG data from 30 to 60 seconds, and so on.

For this part, we provide you with a jupyter notebook (Technical test - Template.ipynb) which contains a few helper functions.
Q1: Plot the EEG and accelerometer for a few nights. What do you see?
Q2: Following your observation, what kind of preprocessing would you advise?
Q3: Train and evaluate a sleep staging model using the provided data. You can use the Deep Learning or Machine Learning framework of your choice.
Q4: How could you summarize the performance of your model?

Part 2 - Serving the model for a frontend web app:

Description: In order to display the hypnogram of a night to the end-user, a web app has been built by the Front-End team.
They need to access the hypnograms for a night through a rest API using a GET request.
In this part, you are asked to build a small python backend (for instance using Flask or Django), which can compute the hypnogram for the specified record and return it.

For each record, there is an associated h5 file stored on this AWS bucket:
https://dreem-ml-case-study.s3.eu-central-1.amazonaws.com/index.html
The h5 associated with the record **87748119-6fff-45d2-9219-888532fb7efd** is *87748119-6fff-45d2-9219-888532fb7efd.h5*
We provide an example to list and download the h5 from the bucket in the *download.py* file.

Specification :
GET /hypnogram/{record_identifier}
If the h5 file exists for this record: compute and return the hypnogram with the following format [“WAKE”, “WAKE”, “N1”,”N1”, “N2”, “DEEP”, ….]
Else: return a code 404
Q1. Implement the backend using a python framework of your choice