|  |  |  |
| --- | --- | --- |
| **Spring 2024** | **Report #2– 04/14/2024** | **Dingyi Nie** |

**——————————————————————————————————————**

**I. Task Achieved Last Week**

**——————————————————————————————————————**

* Re-examined prepared datasets, looked into each feature column’s statistics (missing rates specifically), and enabled a pre-imputation feature selection stage that drops features above some missing rate threshold.
* Implemented the progressive XGBoost imputation strategy.

**——————————————————————————————————————**

**II. Feedback and Interaction**

**——————————————————————————————————————**

* **Prof. Kuo’s Feedback**
* Ignore features with high missing rates for the time being. If they’re really important for the downstream task, assume they are constant
* First run XGBoost: using interpolation (linear / spline); second run XGBoost: using corrected values from first run XGBoost; compare differences; **will it converge if we do more runs?**
* How number of iterations would effect downstream task performance

**——————————————————————————————————————**

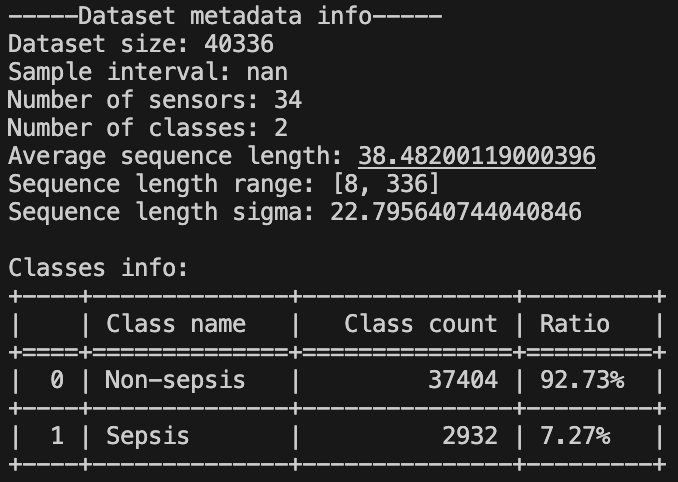
**III. Report**

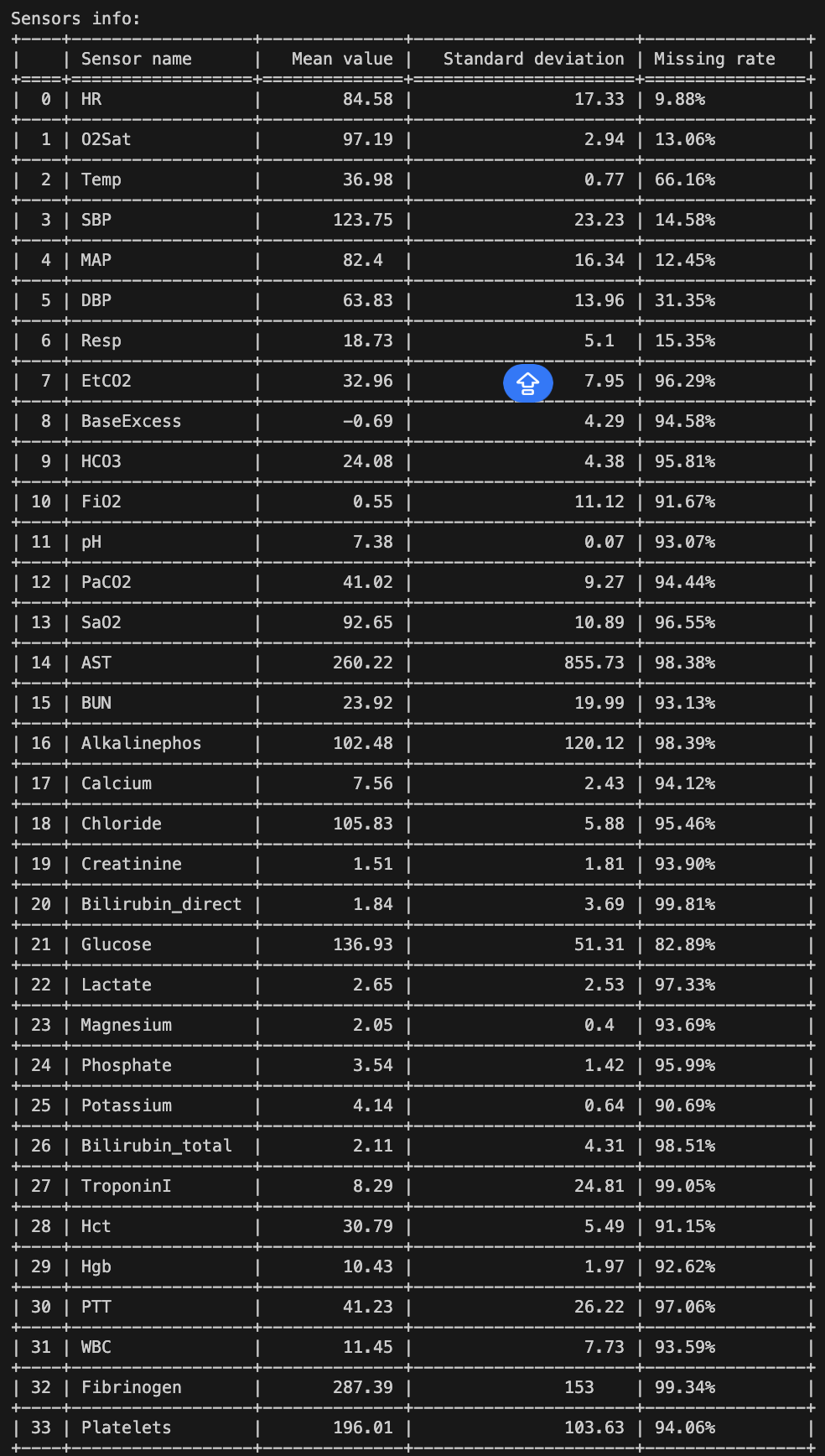
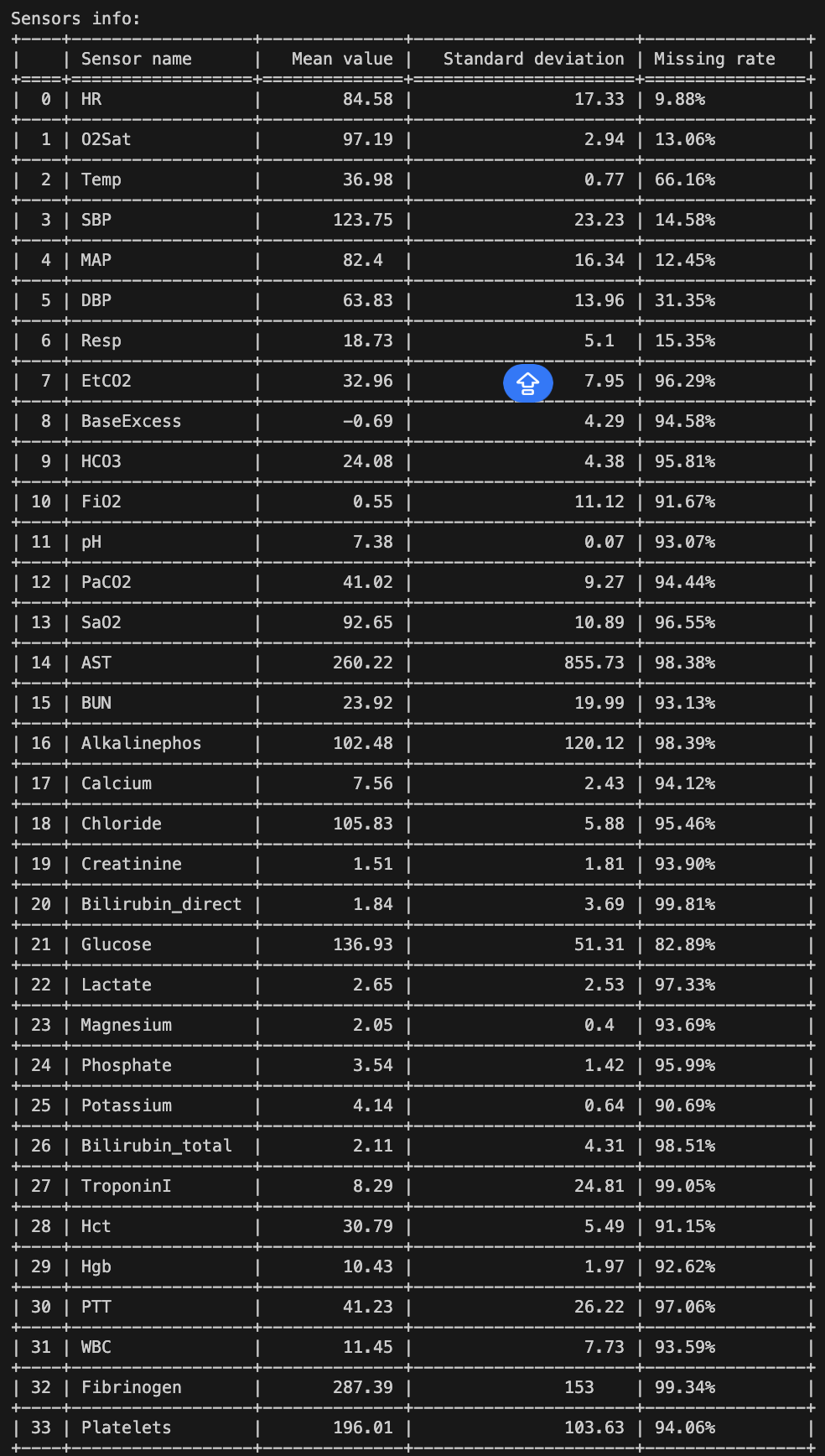
**——————————————————————————————————————**

As discussed last week, I move on with the XGBoost-based imputation strategy this week.

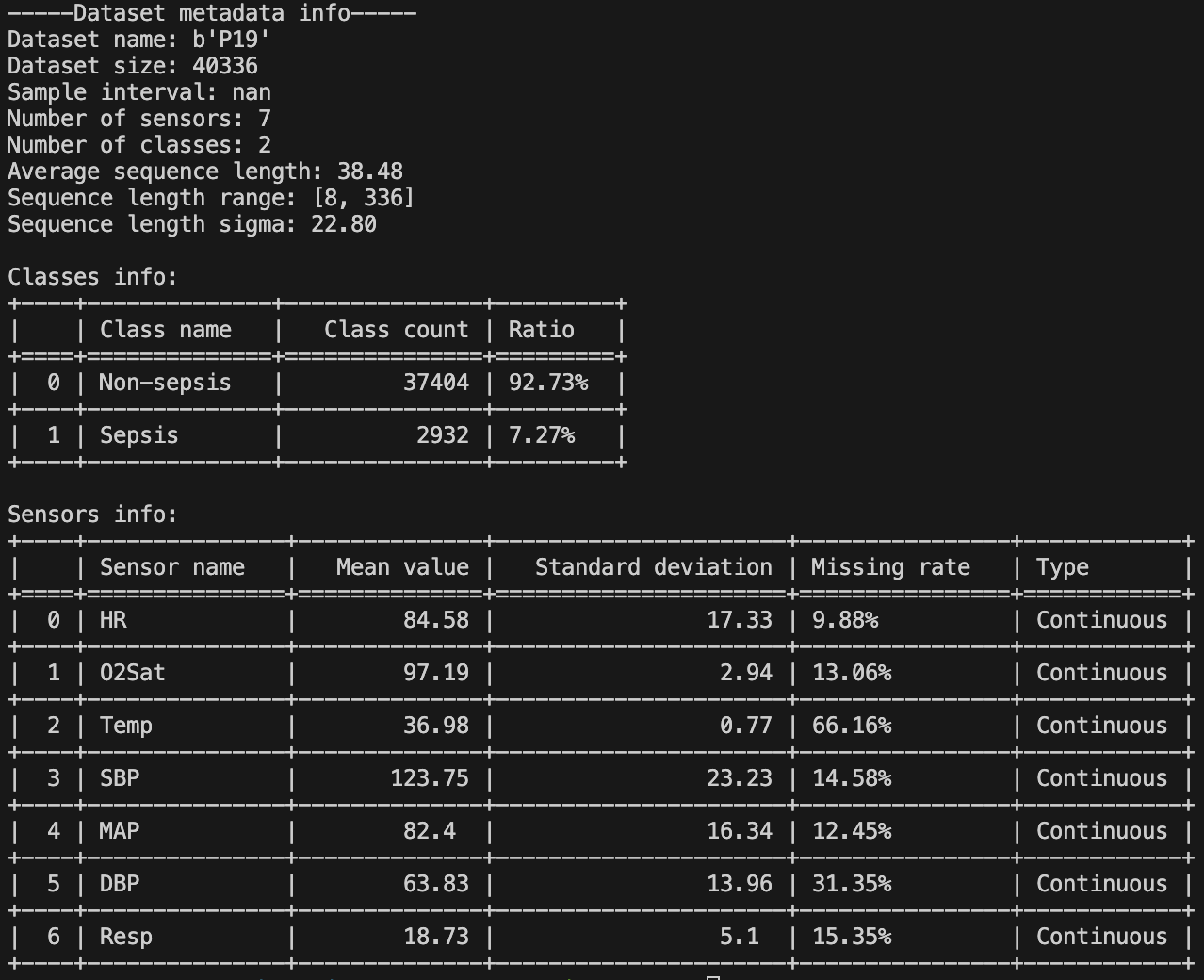
As the dataset of current focus – P19 – has a relatively large number of sensors (34 in total), I select features (sensors) with a missing rate less than 70% only, and fit an XGBoost model for each feature sensor progressively, from the most complete ones to the most missing ones. Also, P19 has a small average sample length (38), so fitting an independent set of XGBoost models on each data sample (episode) would be impossible. These models are thus fit to the whole dataset, meaning that we’re assuming no difference between each patient’s data correlation patterns.

Detailed statistics of P19 dataset:

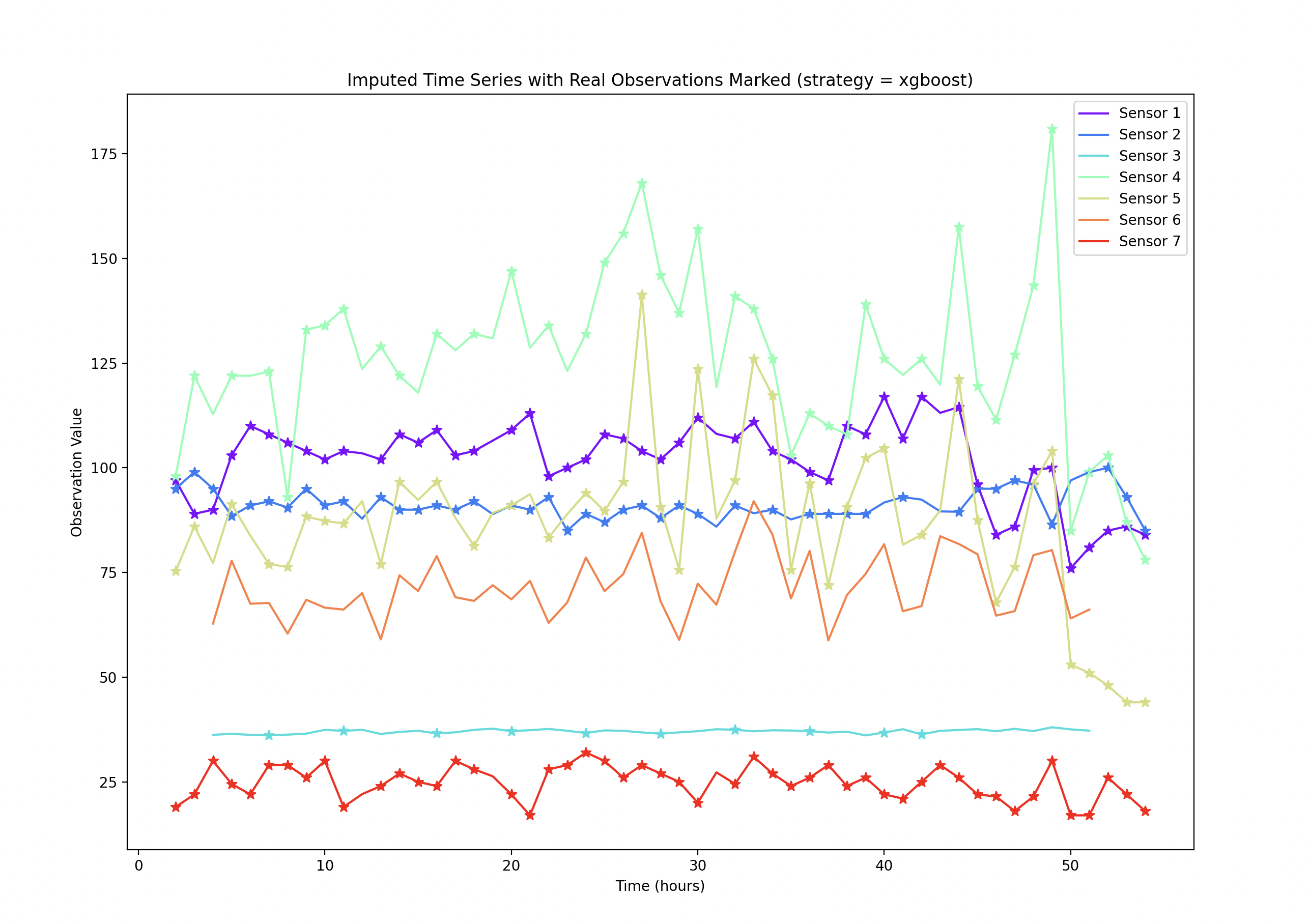




After imputation:



Imputation result visualized example:



Codes are available here: <https://github.com/d9sus4/GL-TS>

**——————————————————————————————————————**

**IV. Next Steps**

——————————————————————————————————————

* Implement downstream classification module with Green Learning tools (already started this week, learning previous codes takes time).
* Complete the full pipeline of Green Learning for modeling irregular time series with missingness.
* Test the classification performance against competitors.

**——————————————————————————————————————**

**V. Milestone**

——————————————————————————————————————

* As stated in I.