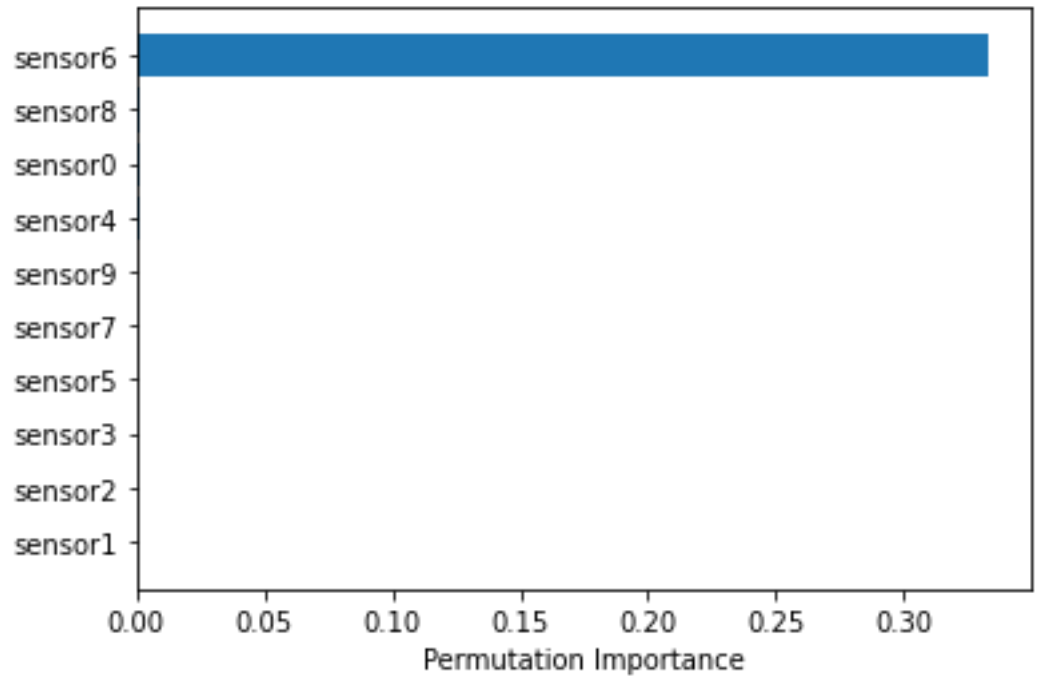
* your process of thought, i.e., how did you come to your solution?

First, I used one of my favorite tree-based method, XGBoost, and I got perfect classification. As I removed the input variables that are least important based on feature importance metrics called permutation importance [1], I end up one variable (sensor6) and it produced perfect classification as well. Thus, other than sensor6 variable, all other input variables are not necessary.



* properties of the artificially generated data set

Data is normalized to 0-1 range as features has minimum values of close to 0 and maximum value of close to 1.

Normalized Value =
(Data-Minimum of Variable i)/(Maximum of Variable i-Minimum of Variable i)

* strengths of your method: why does it produce a reasonable result?

XGBoost is an ensemble method as it aggregate results of many small size trees[5]. Thus, it is more likely to give higher accuracy compared to other methods. Furthermore, it captures non linearity at the data as it is tree-based method.

* weaknesses of your method: when would the method produce inaccurate results?

Small change in the data may result in big change in the decision trees structure and XGBoost is sensitive to outliers[4] This may produce inaccurate results. Furthermore, it may overfit the data If the size of the data is small[2]. In addition, it is not performing well in time series data [3].

*scalability of your method with respect to number of features and/or samples.

XGBoost is an ensemble method. It produces lots of small size decision trees in a systematic way and tree sizes does not grow with features and /or samples[5]. Thus, it is scalable as number of features and/or samples increases.

* alternative methods and their respective strengths, weaknesses, scalability

All the tree-based gradient boosting methods such as Light GBM, CatBoost can be alternative methods. It behaves same as XGBoost in terms of strength, weakness and scalability. XGBost might take more time to train and CatBoost may provide better results when input variables are categorical.

Reference

[1]**https://mljar.com/blog/feature-importance-in-random-forest/**

[2] https://www.kaggle.com/questions-and-answers/77947
[3]https://towardsdatascience.com/why-xgboost-cant-solve-all-your-problems-b5003a62d12a
[4] https://www.quora.com/Is-XGBoost-robust-to-outliers

[5]https://datascience.foundation/datatalk/xgboost-an-efficient-implementation-of-gradient-boosting