**Report 4**

**LI, Ruizhe | 1155076990**

**Github:** <https://github.com/rzli6/ML-Storage.git> (private)

**Target Paper:** Predicting Disk Replacement towards Reliable Data Centers

**Goal:** Realize the method descripted in the paper.

In this week, I applied the method to every model described in the paper, to see whether this predicting model is equally applicable to other disk models. The followings are the result:

1. **Overview**

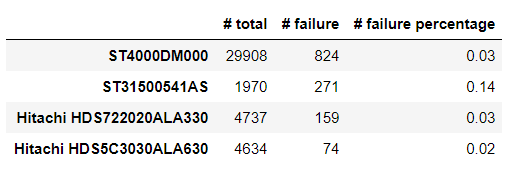


Figure 1: the overview statistics of each disk model

Before tapping into the training procedure and predicting result, we should first take a look at the data set we are currently working on. Based on the figure 1 shown above, the ST4000DM000 (Sgt A) model has the biggest total number of disks, and it has a failure proportion of 3%. And following is the Hitachi HDS722020ALA330 (Hit A), which has 4737 disks in total, and 159 failed disks. Keeping this information in mind could help us later construe the discrepancies among different predicting results when applying the same method on dissimilar models.

1. **Apply to different model**

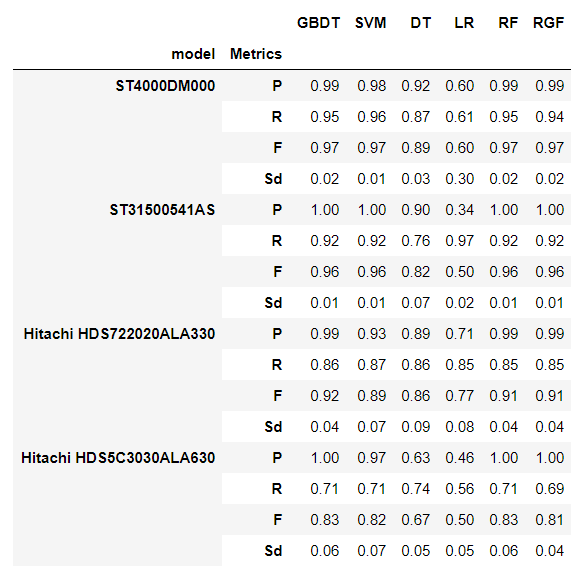


Figure 2: the precision, recall, f1 score, and Sd of each model, obtained from experiment

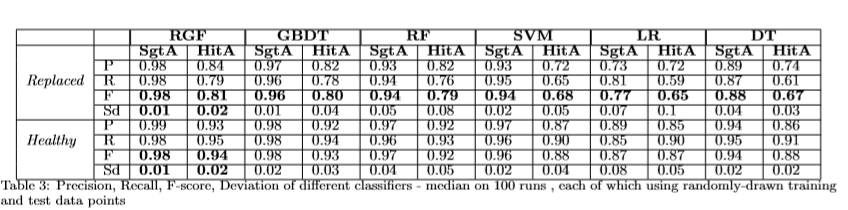


Figure 3: the precision, recall, f1 score, and Sd of each model, stated in the paper

When I tried to apply the same machine-learning method on different disk models, the result is shown in Figure 2. Basically, this method can reach a predicting accuracy like 80% high. And different machine learning methods actually got different result. For example, in general the predicting accuracy of LR (Logistic Regression) model is way less than the RGF model, which is the most preferred method according to the paper author.

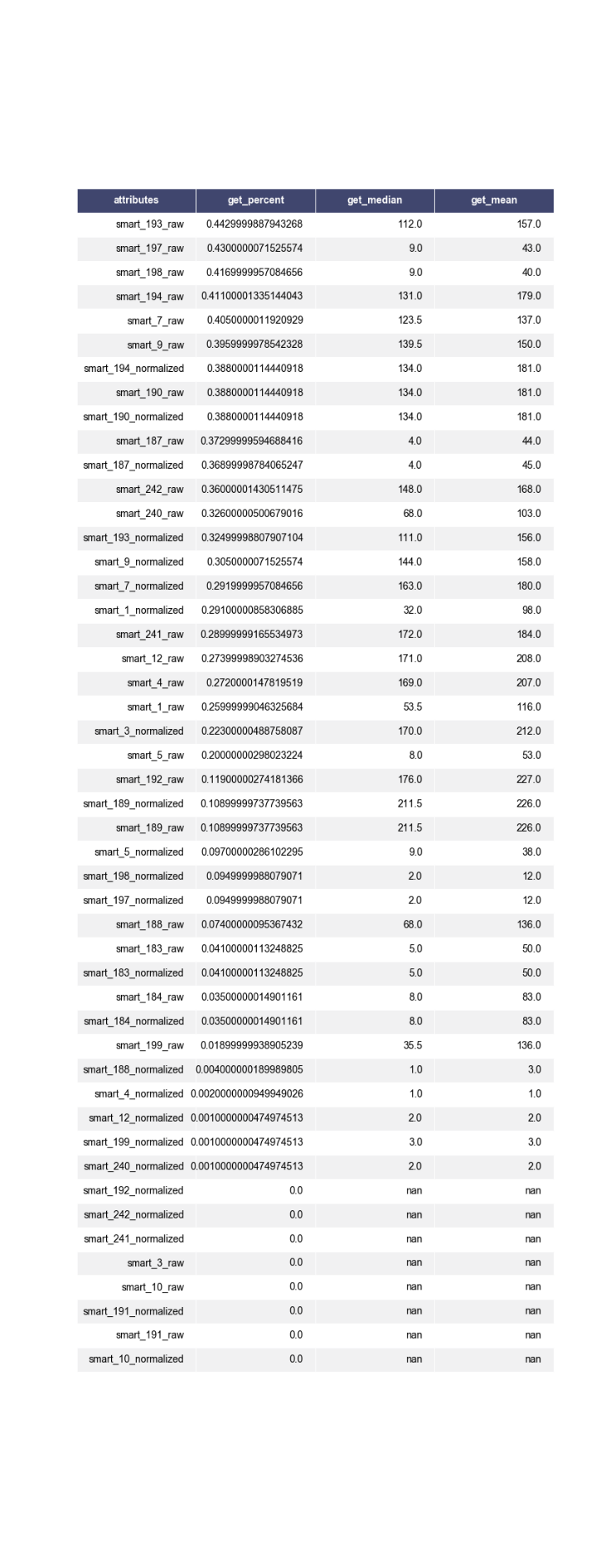
2.1. Similarities

My result has several similarities comparing to the one got from paper. Take Sgt A as an example. The orders of accuracy of different machine learning model are **similar:** **RGF >= GBDT >= RF >= SVM >= DT >= LR.** These consistency means RGF, GBDT, RF models do have some strengths on answering this kind of questions. The algorithms behind “tree structure” can properly interpret relationship between health status and the selected SMART attributes of each disk. And the result also indicates some differences among the accuracies of several disk models. If we focus on the RGF model, order of accuracies of each model is Sgt A > Sgt B > Hit A > Hit B, which is consistent with the descending order of the number of failed disks, and this order seems not related to the total number of disks and failed percentage. Down sampling could be one of key reasons for this relationship. In the experiment, the data is sampled to be commensurate to failed number. Therefore, the size of training set is not determined by the whole data set, but the number of failed disks.

* 1. Differences

However, there are also some discrepancies. For example, in the paper, RGF is a dominant machine learning method. It shows a conspicuous advantage over other method. On the contrary, this huge gap did not show up in my result. GBDT, SVM, RF, and RGF can all reach a F1 score around 97%, in Sgt A model. This could be interpreted with different attributes we selected. I selected 6 more attributes than the paper. And based on my understanding, why these attributes were not shown in the paper is because either some attributes were not accessible at that time, or the author thought they were irrelevant. And the bigger number of attributes may lead to the differences in which model my accuracy is higher. In addition to attributes, another difference between the author and I is our datasets was not completely same. He used a larger dataset, from 2013 to 2015 excluding several months in between, totally 27 months’ data. However, my dataset only contained the data from 2015 whole year, 12 months’ data. Thus, the accuracy dropped in my result could be explained by the decrement of dataset size.

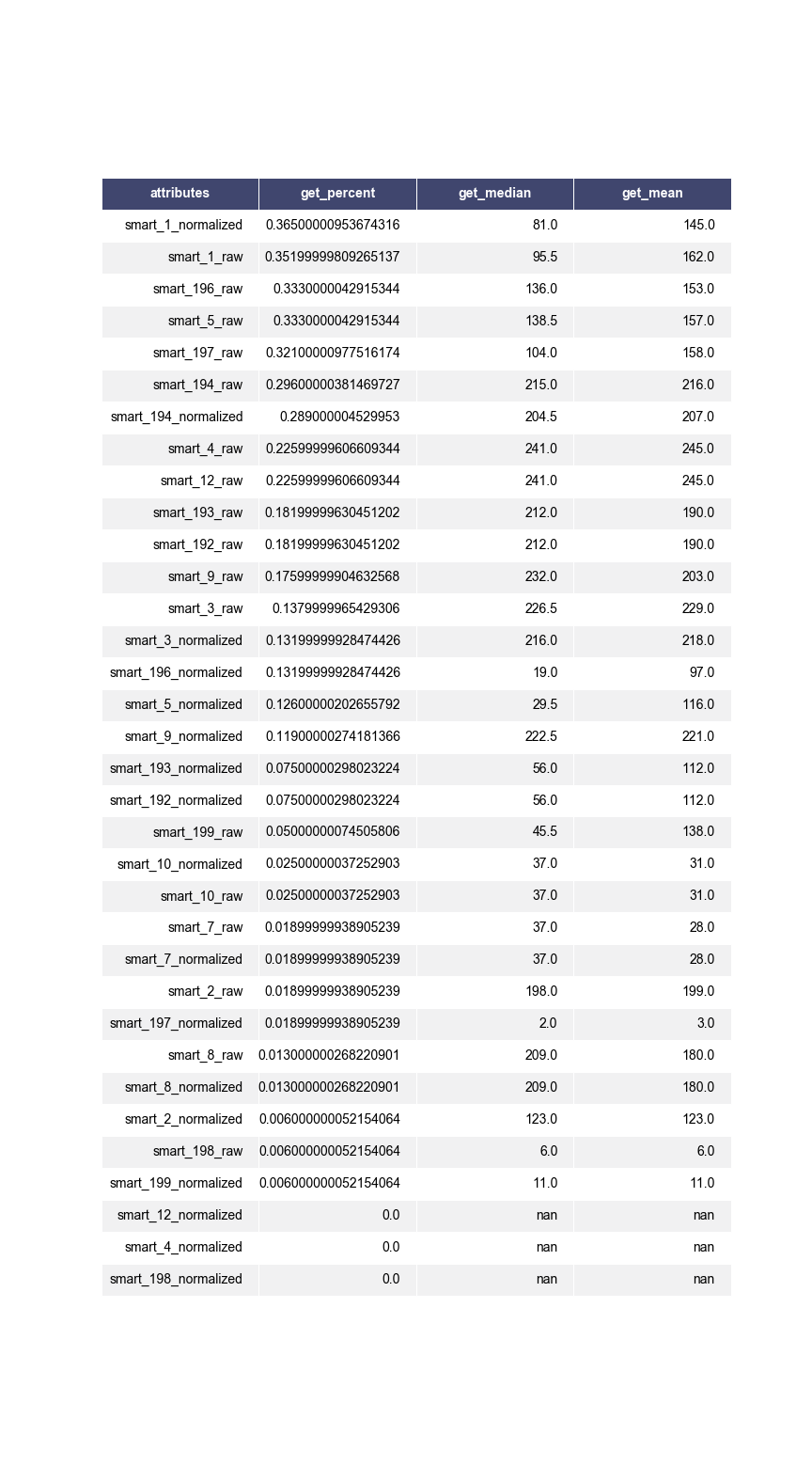
1. **Feature summarize of each model.**



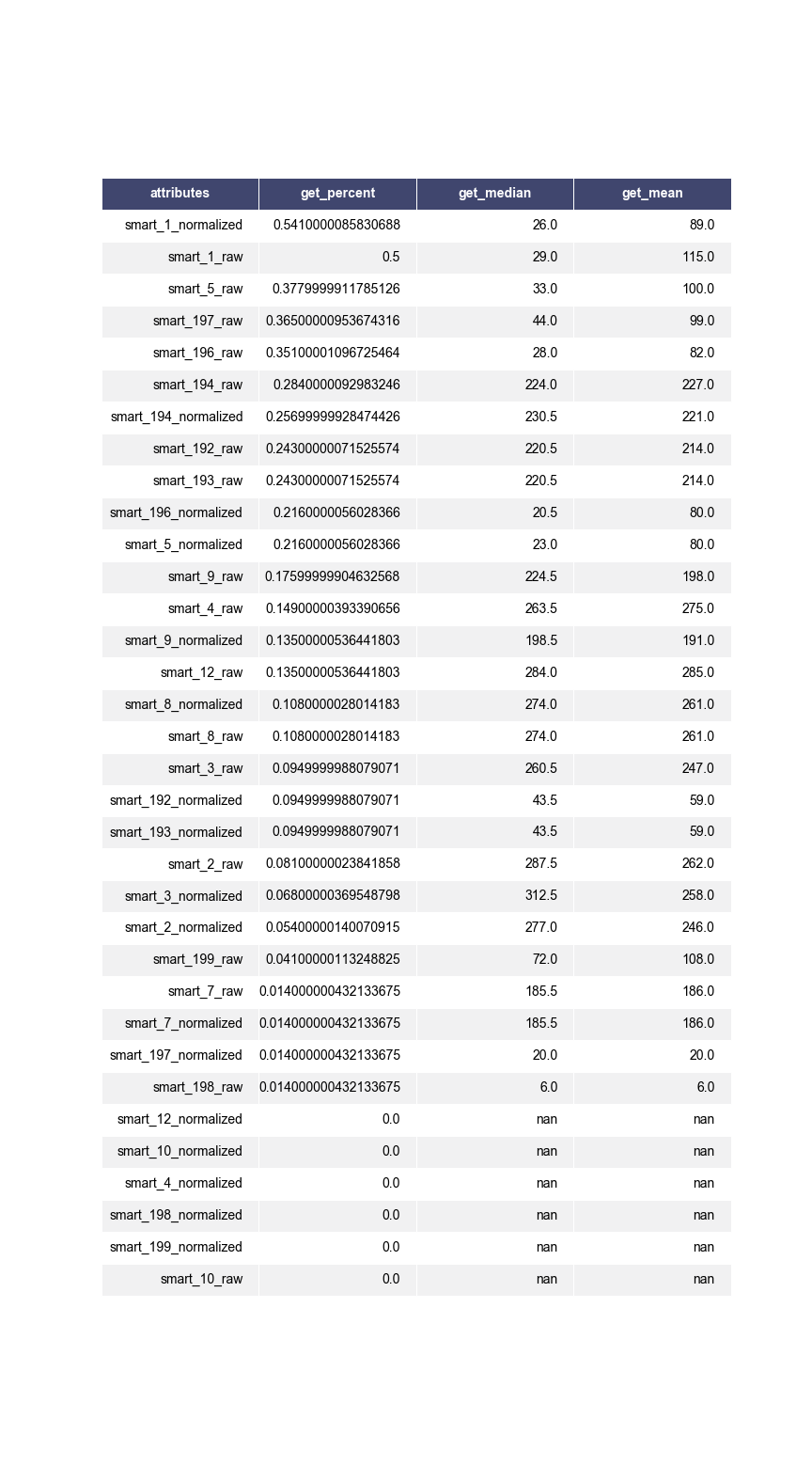
Sgt A



Sgt B



Hit A



Hit B