WEEKLY REPORT 3

YANG zhi lin - 19-06-2018

1. **Data preprocessing**:

Model: Seagate A 'ST4000DM000'
Date range: 2017-01-01 ~ 2017-12-31
Number of samples: 35189

1) **Feature selection**:
After discarding the features with not enough records, there are **48** available features in total:

```
[‘smart_1_normalized', 'smart_1_raw',
'smart_3_normalized', 'smart_3_raw',
'smart_4_normalized', 'smart_4_raw',
'smart_5_normalized', 'smart_5_raw',
'smart_7_normalized', 'smart_7_raw',
'smart_9_normalized', 'smart_9_raw',
'smart_10_normalized', 'smart_10_raw',
'smart_12_normalized', 'smart_12_raw',
'smart_183_normalized', 'smart_183_raw',
'smart_184_normalized', 'smart_184_raw',
'smart_187_normalized', 'smart_187_raw',
'smart_188_normalized', 'smart_188_raw',
'smart_189_normalized', 'smart_189_raw',
'smart_190_normalized', 'smart_190_raw',
'smart_191_normalized', 'smart_191_raw',
'smart_192_normalized', 'smart_192_raw',
'smart_193_normalized', 'smart_193_raw',
'smart_194_normalized', 'smart_194_raw',
'smart_197_normalized', 'smart_197_raw',
'smart_198_normalized', 'smart_198_raw',
'smart_199_normalized', 'smart_199_raw',
```

```
'smart_240_normalized', 'smart_240_raw',
'smart_241_normalized', 'smart_241_raw',
'smart_242_normalized', 'smart_242_raw']
```

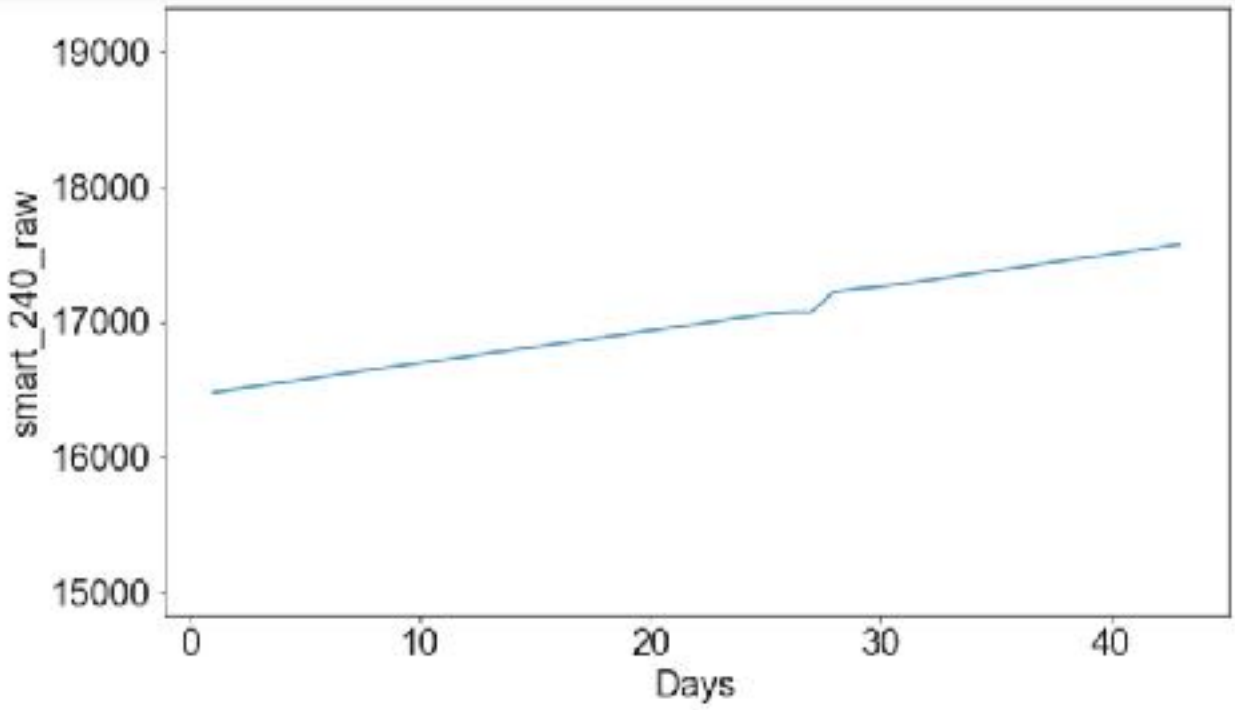
2) **failed samples collection**:
1061 replaced disks among **35189** samples

3) **change point detection on failed samples**:
Method: CPD frequentist approach

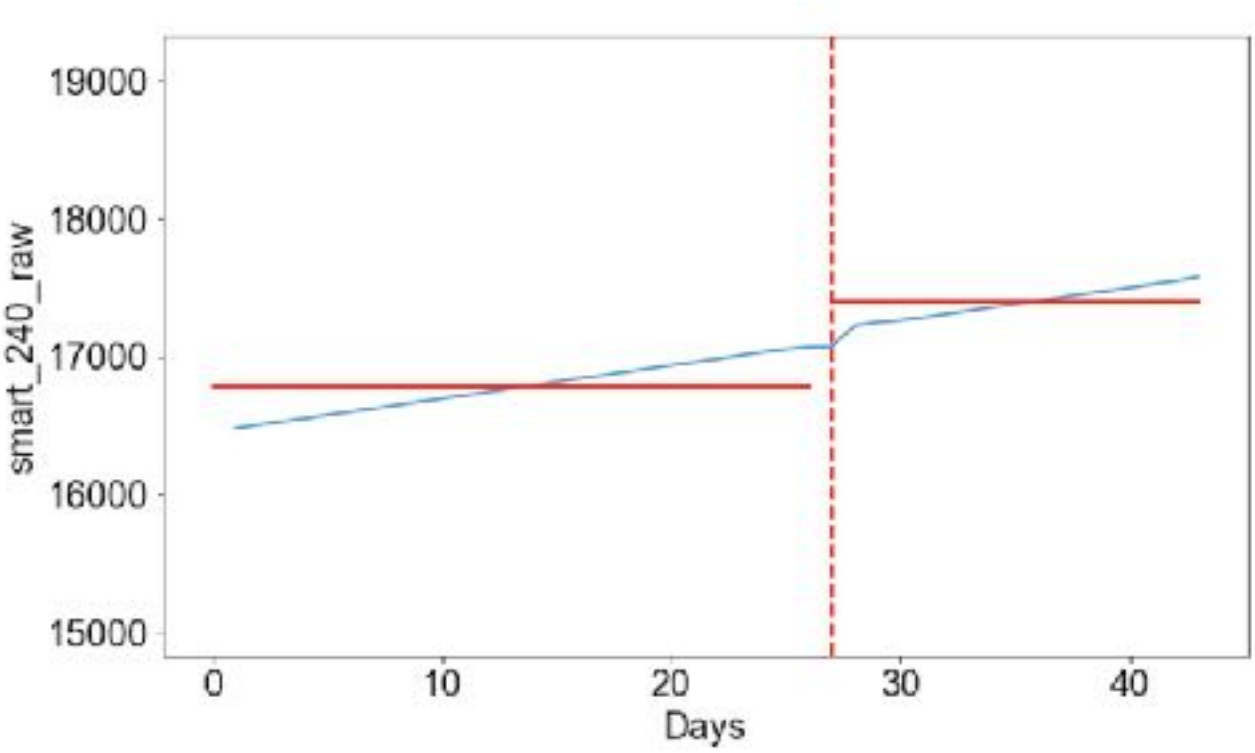
Code Source: http://www.claudiobellei.com/2016/11/15/changepoint-frequentist/

Model: **Normal distribution, single change point, change in mean**

Example: Take sample 'Z3029FAS'




```
----------------------
2G = 7.331324e+06
sigma**2*lambda = 1.993853e+05
-->H0 rejected
Changepoint detected at position: 27
m1 = 16707.518519
m2 = 17391.562500
```



As the figures show, this attribute actually has a significant shift on day 27 and it's captured by the program. But here is another problem: it seems that it is not a permanent change, but rather more like a temporary change? Is it really indicative of a failure?

After change point detection, I have aggregated the distribution of days before replacement of each SMART attributes. The median values have been used as the window width in exponential smoothing:

	smart_1_normalized	smart_1_raw	smart_3_normalized	smart_3_raw
get_median	24	49	101	
get_mean	68	85	113	

smart_4_normalized	smart_4_raw	smart_5_normalized	smart_5_raw	smart_7_normalized
	86	4	4	94
	100	16	17	106

smart_12_normalized	smart_12_raw	smart_183_normalized	smart_183_raw	smart_184_normalized
	86	2	2	1
	100	25	25	10

smart_189_normalized	smart_189_raw	smart_190_normalized	smart_190_raw	smart_191_normalized
115	115	90	91	
133	132	111	111	

smart_191_raw	smart_192_normalized	smart_192_raw	smart_193_normalized	smart_193_raw
		68	72	72
		100	90	91

smart_194_normalized	smart_194_raw	smart_197_normalized	smart_197_raw	smart_198_normalized
91	91	2	6	2
111	111	4	22	4

smart_198_raw	smart_199_normalized	smart_199_raw	smart_240_normalized	smart_240_raw
6		39		86
22		95		88

smart_7_raw	smart_9_normalized	smart_9_raw	smart_10_normalized	smart_10_raw
74	89	86		
85	91	88		

smart_241_normalized	smart_241_raw	smart_242_normalized	smart_242_raw
	103		67
	104		80

smart_184_raw	smart_187_normalized	smart_187_raw	smart_188_normalized	smart_188_raw
1	4	4		5
10	26	26		47

Then I count the SMART correlation frequencies for those features with change points:

(correlation frequency: the percentage of drives for which a correlation with disk is observed)

Correlation frequencies

	percent
smart_1_normalized	30.62%
smart_1_raw	24.57%
smart_3_normalized	14.65%
smart_4_raw	33.08%
smart_5_normalized	6.14%
smart_5_raw	20.51%
smart_7_normalized	41.78%
smart_7_raw	61.91%
smart_9_normalized	54.25%
smart_9_raw	58.79%
smart_12_raw	33.08%
smart_183_normalized	12.76%
smart_183_raw	12.76%
smart_184_normalized	2.08%
smart_184_raw	2.08%
smart_187_normalized	33.93%
smart_187_raw	33.93%
smart_188_raw	0.85%
smart_189_normalized	2.36%
smart_189_raw	2.46%
smart_190_normalized	54.06%
smart_190_raw	53.97%
smart_192_raw	5.01%
smart_193_normalized	30.06%
smart_193_raw	58.32%
smart_194_normalized	53.97%
smart_194_raw	53.97%
smart_197_normalized	9.74%
smart_197_raw	44.71%
smart_198_normalized	9.74%
smart_198_raw	44.71%
smart_199_raw	0.38%
smart_240_raw	58.32%
smart_241_raw	48.02%
smart_242_raw	65.12%

Those red marked features are the features I selected to train the ML classifiers:

Totally 25 features.

4) **Exponential smoothing**:

Method: ewma in pandas library

After 4 stages of data preprocessing, I collected 15001 samples in total, with 458 replaced disk samples.

2. **Downsampling on healthy samples**:

Method: K-Means clustering

totally: **1000** healthy samples and **458** replaced samples to feed into ML classifiers

3. **Training Machine Learning Classifiers**:

Basic Method: 10-folds cross validation

Healthy

	RGF	GBDT	RF	SVM	LR	DT
Precision	0.916 +/- 0.018	0.907 +/- 0.012	0.913 +/- 0.015	0.686 +/- 0.002	0.686 +/- 0.002	0.925 +/- 0.013
Recall	0.985 +/- 0.010	0.981 +/- 0.012	0.974 +/- 0.014	1.000 +/- 0.000	1.000 +/- 0.000	0.916 +/- 0.031
F-score	**0.949 +/- 0.009**	0.942 +/- 0.009	0.942 +/- 0.010	0.814 +/- 0.001	0.814 +/- 0.001	0.920 +/- 0.017

Replaced

	RGF	GBDT	RF	SVM	LR	DT
Precision	0.962 +/- 0.026	0.950 +/- 0.029	0.935 +/- 0.035	0.000 +/- 0.000	0.000 +/- 0.000	0.824 +/- 0.054
Recall	0.801 +/- 0.046	0.780 +/- 0.031	0.797 +/- 0.036	0.000 +/- 0.000	0.000 +/- 0.000	0.838 +/- 0.032
F-score	**0.873 +/- 0.026**	0.856 +/- 0.022	0.860 +/- 0.024	0.000 +/- 0.000	0.000 +/- 0.000	0.830 +/- 0.032

The result in the paper:

		RGF		GBDT		RF		SVM		LR		DT	
		SgtA	HitA	SgtA	HitA	SgtA	HitA	SgtA	HitA	SgtA	HitA	SgtA	HitA
Replaced	P	0.98	0.84	0.97	0.82	0.93	0.82	0.93	0.72	0.73	0.72	0.89	0.74
	R	0.98	0.79	0.95	0.78	0.94	0.76	0.95	0.05	0.81	0.58	0.87	0.61
	F	**0.98**	**0.81**	**0.96**	**0.80**	**0.94**	**0.79**	**0.94**	**0.68**	**0.77**	**0.65**	**0.88**	**0.67**
	Sd	0.01	0.02	0.01	0.04	0.05	0.08	0.02	0.05	0.07	0.1	0.04	0.03
Healthy	P	0.98	0.95	0.93	0.92	0.97	0.92	0.97	0.87	0.88	0.85	0.94	0.86
	R	0.98	0.95	0.93	0.94	0.96	0.93	0.96	0.90	0.85	0.90	0.95	0.91
	F	**0.98**	**0.94**	0.93	0.93	0.97	0.92	0.96	0.88	0.87	0.87	0.94	0.88
	Sd	0.01	0.02	0.02	0.03	0.04	0.05	0.02	0.04	0.05	0.05	0.02	0.02

Table 3: Precision, Recall, F-score, Deviation of different classifiers - median on 100 runs, each of which using randomly-drawn training and test data points

Analysis:

<1> Same as what in the paper, **RGF** achieved the highest F-score among all classifiers on both healthy disks and replaced disks.

<2> All **tree based machine learning methods** get fairly good performance.

<3> The F-scores of replaced disks are generally lower than those of healthy disks. These are mainly due to the class imbalance.

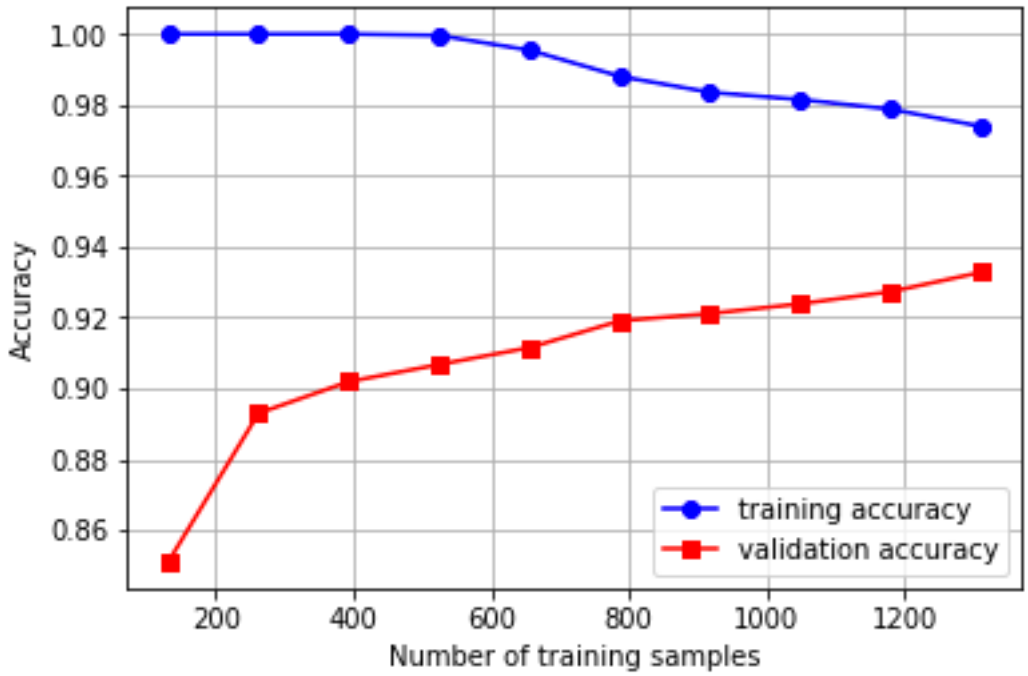
<4> The **zero precision problem** still exists for LR classifier and SVM classifier. When I actually checked their prediction results, I find they actually managed to predict several replaced disks but the ratio is just too low. I think it may be because these two classifiers are more sensitive to class imbalance.

4. **Assessment:**

(1) **Learning curves of RGF**:

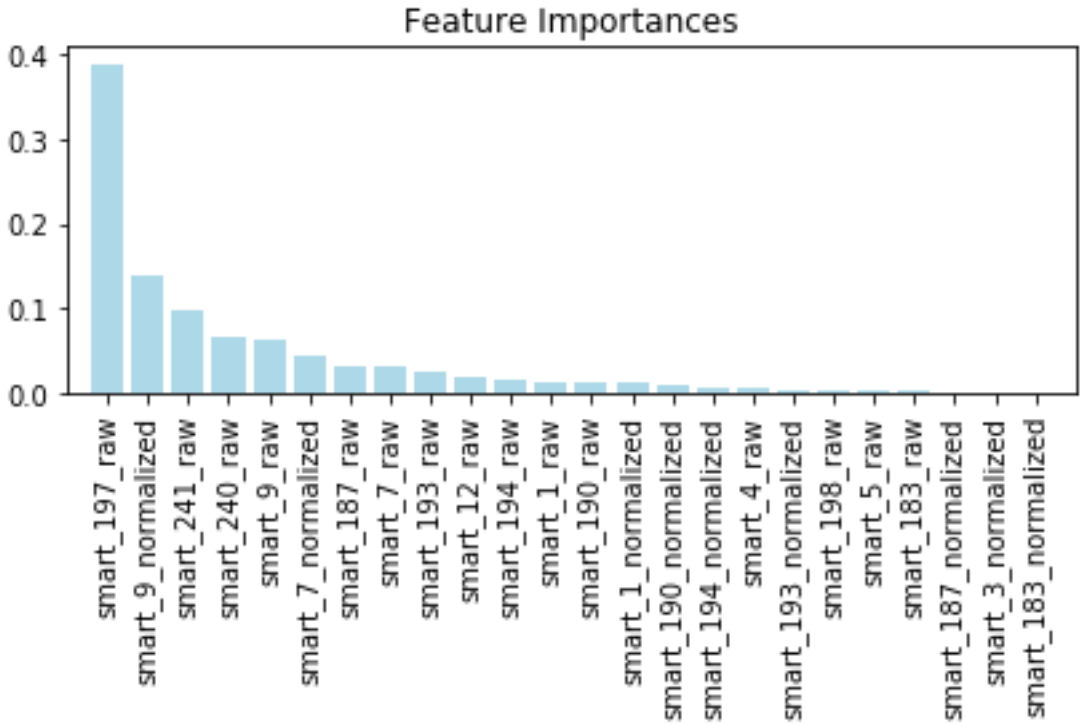
Analysis:

1. Both accuracies are reasonably high, which shows that the learning result has a good bias-variance trade-off.
2. The validation accuracy shows a tendency of increase, which suggests the potential of further improvement by collecting more training samples.



(2) **Feature importance assessment**

Method: Measure feature importance as **the averaged impurity decrease** computed from the decision tree



5. Generate SMART indicator rules by the decision tree:

Method: Sketch the decision tree and trace its decision process

Image: Github link: Machine-Learning-Research-Project/decision_tree.pdf

My problem:

1. The paper doesn't give details on how to actually generate the indicator rules and how to measure the confidence rate?
2. Since different decision trees split data space in different ways (different choices of features and different threshold), the indicator rules generated by different classifiers will be quite different.

6. Transfer learning

(1) **Model:** Seagate A and B
Time: 2015 whole year
Sample Statistics:

Seagate A	15001 disks in total	572 replaced disks
Seagate B	1693 disks in total	109 replaced disks

After **downsampling** SeagateA:
Seagate A 1572 disks in total 572 replaced disks

(2) Prediction result of SgtB failure **before transfer learning**:
Classifier1: RGF training on SgtA
Precision: 0.362
Recall: 0.899
F1: 0.516

(3) Cross validation result for **classifying A and B**:
Classifier2: RGF training on the combined set of SgtA and B
precision: 0.875 +/- 0.020
recall: 0.968 +/- 0.007
f1: 0.919 +/- 0.010

(4) **Select samples from SgtA** that are representative of B:
Classifier: Classifier2 in step (3)
Result: Select 1690 SgtA samples in total (107 replaced disks)
(Fortunately, this size is comparable to SgtB)

(5) Prediction result of SgtB **after transfer learning**:
Classifier: Classifier3 trained on SgtA dataset from step (4)
Precision: 1.000
Recall: 0.725
F1: 0.840