Zachari THIRY
Benjamin SALON
Matthieu RAMON

Machine learning in industry: Final deliverable

Abstract _ Our goal is to predict maintenance needs for NASA turbofans using Machine Learning (ML) techniques, in a practical case.

INTRODUCTION:

Predicting machine failure is a big and rewarding challenge for industrial companies. Nowadays, most modern and high-tech machines have sensors able to record features such as machine temperature, rotation speed, or vibration. These features can be used by predictive maintenance analysts to predict wear and schedule maintenance.

The dataset we used is different from the previous deliverables. The former was artificially and randomly generated, and we rather work on a more concrete real life example. The latter was generated over NASA turbofan engine tests.

THE DATA :

1. DATA PRESENTATION

The dataset consists of 160 000 points each containing 33 features. It is divided as such :

- Engine number
- Current lifetime in cycles
- 3 features of operation settings
- 27 features of sensor data
- Computed RUL (Remaining Useful Lifetime)

2. DATASET TRANSFORMATION

We first removed the sensors features 22 to 27 as they were empty. Then the scatter matrix showed that 12 other sensors do not provide any significant insight (almost constant w.r.t RUL). As such, they

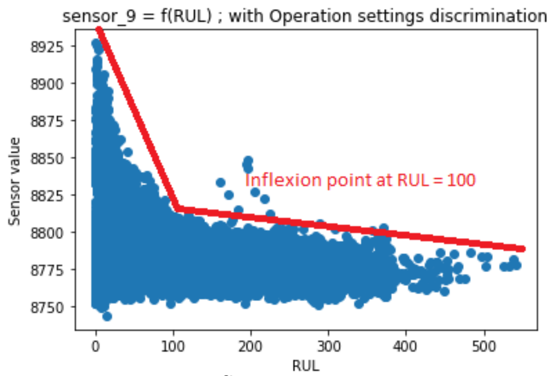


Figure Data.1 : Data inflexion point

were also removed. Also, engine_no was dropped as its predictive power was null.

With the scatter matrix, we saw that sensor data generally showed higher variance for RUL < 100 cycles (see figure Data.1). We thus decided to transform RUL prediction as a classification problem: "Will our engine fail within the next 100 cycles? (yes : label = 1)"

Then, we separated our data in 5 interpretable subsets (see fig Data.2)

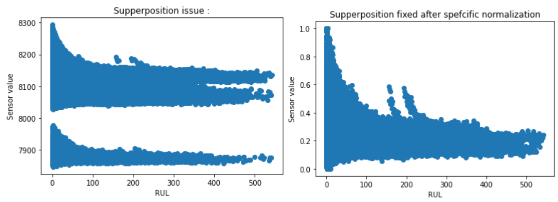


Figure Data.2 – Before – after transformation (sensor 17)

This is one sensor but for all the sensors, we can see that there are up to five different data groups supperposed to each other. We found out that it was related to the five values of op_setting_1.
We thus isolated 5 distinct datasets, normalized them and merged them back together to obtain interpretable dataset.

TRAINING AND PERFORMANCE

All the models will be trained on the normalized and merged dataset previously described in part I.2.

Since our dataset is unbalanced, with the $label = 0$ class being 4 times as dominant, we decided to use the following scoring functions :

- The default score (sklearn default method)
- The f1-score classically used on unbalanced datasets.

1. LOGISTIC REGRESSTION

The first model we trained was a simple logistic regression with default settings. We wanted to have an idea of how a naïve model would perform, establishing a basis for improvement.

Accuracy with such settings was :

$$F1\_score = 82.9\%$$

Varying some hyperparameters, we were unable to improve the results. See figure Training.1 with regularization parameter “C” .

C	Default score	F1_score
1	0.851	0.829
100	0.851	0.829
1000	0.851	0.829

Table Training.1 - LR results 1

2. SVM

Keeping with linear classifiers, we fitted a SVM. Varying the hyperparameters, we weren’t able to yield better results than the logistic regression.

Results were : default score = 85.0%

We tried a RBF kernel to improve the results without success. The default score dropped to 55.1%

3. Trees

Lastly, we opted for decision trees. Althouh this model isn't linear it is widely used in predictive maintenance. In fact, most machines fail after a certain safety criterion is reached, which corresponds to a threshold identifyable by trees.

Our parameter evaluation showed that, the optimum tree depth is 7. Beyond this value, the tree keeps getting deeper but overfits and test results start decreasing (figure Training.2).

In this figure, we introduce :

- Hihg_RUL score (accuracy on the true negatives)
- Low_RUL score (accuracy on the true positives)

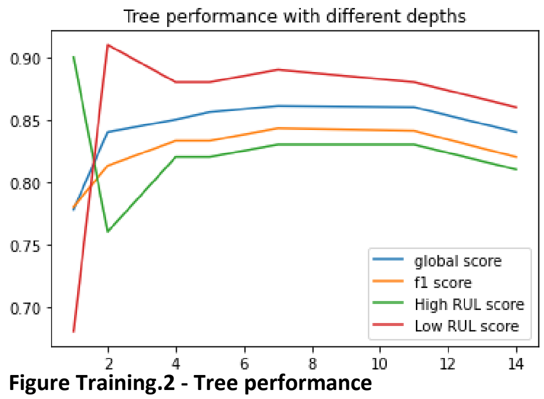


Figure Training.2 - Tree performance

Note that the tree outperforms all other models we tested with the following scores : 86% for global prediction, 83% for low RULs and 89% for the high RULs. It scores beter in these 3 categories.

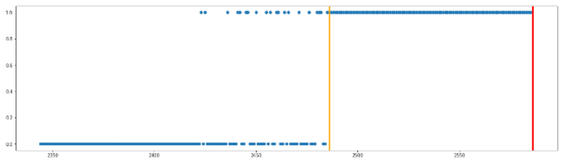
Global Prediction	F1-score	High_RUL	Low_RUL
86.1%	84.3%	83%	89%

ADDITIONAL LAYER FOR LIVE PREDICTION

Median Filtering and Consecutive test

There are two important things for a predictive maintenance model. The first is that the model warns the maintainer that the engine will break before it actually breaks. The second is that the model does not predict a failure too early as it would be too costly: the maintainer would have to change the engine at the firsts failure predictions.

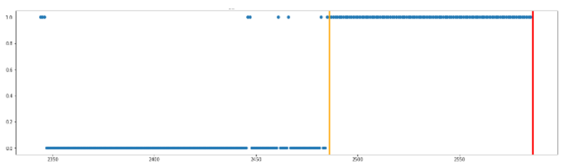
Let’s take an example with a prediction of the random forest model for 1 machine lifetime. In the following pictures we can see for the y axis the prediction of the model (1 if it predicts a failure in the next 100 cycles, 0 if not). The x axis represents the cycles during the lifetime of the machine. The red vertical line corresponds to the cycle where the machine breaks and the yellow one is placed 100 cycles before. In a perfect model, all the predictions before the yellow line would be 0 (in the bottom) and all the predictions between the yellow line and the red line would be 1 (at the top).



1 Prediction for 1 machine lifetime with Random Forest

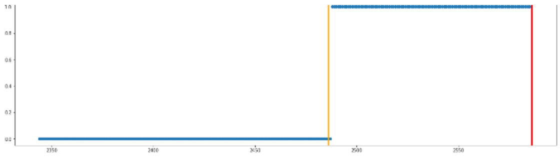
We can see on the figure above that the random forest classifier predict many false positive (all the 1 at the left of the yellow line). As explained we want to prevent this. To do so we used a median filter on the prediction array of the model. The filter looks at the 6 previous predictions and returns the median of the 7 points.

Doing so we obtain this result:



2 - Same prediction after median filtering

We see here that we removed many false positives from the prediction. But this is still not enough. We decided to add another layer: we called it a “consecutive filter”. It looks over the n previous points and returns 1 only if they are all 1. It yields this result:



3 - Same prediction after median filtering and consecutive test (n=4)

We can now see that we have no more false positives and the model is close to a perfect model for this particular lifecycle.

MEMBERS PARTICIPATION

There was no clear distribution of the workload among the three of us. Since we are roommates, we usually worked together on every part of the project.