

# Predictive Maintenance Practice

## A Case of NASA's Aircraft Engines with Data Science Approach

### **Abstract**

This is a machine learning practice case of predictive maintenance by python with NASA's turbofan engine degradation simulation data set. The popular predictive and classification algorithms were implemented to build this project. The codes of this project were deposited on GitHub:

<https://github.com/littleHurt/Predictive-Maintenance-Practice>

Author: Kira Wang

Submit Date: 2019-12-21



Credit by Aerospace Engineering

# Contents

<b>1 Introduction.....</b>	<b>1</b>
1.1 Predictive Maintenance.....	1
1.2 Procedure & Approach.....	1
1.3 Environment .....	1
<b>2 Data Preprocessing .....</b>	<b>2</b>
2.1 Data Acquisition .....	2
2.2 Exploratory Data Analysis & Feature Engineering .....	3
2.2.1 Dimension Reduction .....	4
2.2.2 Feature Construction .....	5
<b>3 Regression Modeling.....</b>	<b>6</b>
3.1 Algorithms & Metrics .....	6
3.2 Analyses of Results.....	7
<b>4 Classification Modeling .....</b>	<b>10</b>
4.1 Algorithms & Metrics .....	10
4.2 Analyses of Results.....	11
4.3 Benefit Evaluation .....	14
<b>5 Conclusions.....</b>	<b>15</b>
5.1 Summary .....	15
5.2 Extension of Future.....	15
<b>References .....</b>	<b>16</b>

# 1 Introduction

## 1.1 Predictive Maintenance

Predictive maintenance (PdM) is a technique that uses condition-monitoring tools and techniques to track the performance of equipment during normal operation to detect possible defects and fix them before they result in failure. Ideally, predictive maintenance allows the maintenance frequency to be as low as possible to prevent unplanned reactive maintenance, without incurring costs associated with doing too much preventive maintenance (PvM).<sup>1</sup> In this project, we want to figure out the answer of two key questions with the issue of predictive maintenance:

1. How many unexploited running cycles do the engines remain?
2. Whether the engines will break in the following cycle?

## 1.2 Procedure & Approach

The project would be implemented under the following procedure:

1. Loading data and understand data [exploratory data analysis](#) (EDA).
2. Implement data wrangling and [feature engineering](#) for modeling.
3. Predict remaining useful lifecycle by [regression algorithms](#).
4. Determine whether the engine will break in this cycle by [binary classification algorithms](#).
5. Summarize the results from machine learning modeling.

## 1.3 Environment

This project was practiced under the environment below:

Table 1 Project practice environment

<b>System</b>	Windows 10	Intel(R) i7-6700 CPU @ 3.40GHz, RAM 8G
<b>Software</b>	Python 3.8	Spyder IDE 4.0 (Anaconda3)
<b>Packages</b>	Data Wrangling	NumPy 1.17 & pandas 0.25
	Visualization	Matplotlib 3.1.0 & seaborn 0.9.0 & Scikit-plot 0.37
	ML algorithm	scikit-learn 0.21.3 & xgboost 1.0

---

<sup>1</sup> 【Predictive Maintenance: What is PdM Maintenance?】 | Fiix  
<https://www.fiixsoftware.com/maintenance-strategies/predictive-maintenance/>

## 2 Data Preprocessing

### 2.1 Data Acquisition

The data we used in this project were the simulated running data of aircraft which provided by NASA.<sup>2</sup> Data sets consists of multiple multivariate time series. Each data set is further divided into training and test subsets. Each time series is from a different engine i.e., the data can be considered from a fleet of engines of the same type.

The engine is operating normally at the start of each time series, and develops a fault at some point during the series. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure. The objective of the competition is to predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate. Also provided a vector of true Remaining Useful Life (RUL) values for the test data.

The training dataset contains 20,631 observations of run-to-failure data of aircraft engines of 100 engines. The test dataset contains 13,096 observations of operation data of 100 aircraft engines without records of failure events. The ground truth dataset contains the true remaining cycles of each engine in the test dataset and saved e in another file

Table 2 Samples of training dataset

id	cycle	st1	st2	st3	s1	s2	...	s20	s21
1	1	-0.0007	-0.0004	100	518.67	641.82	...	39.060	23.419
1	2	0.0019	-0.0003	100	518.67	642.15	...	39.000	23.424
...	...	...	...	...	...	...	...	...	...
100	199	-0.0011	0.0003	100	518.67	643.23	...	38.290	23.064
100	200	-0.0032	-0.0005	100	518.67	643.85	...	38.370	23.052

Table 2 show the a few samples of training dataset, where column “id” denotes the id number of engines; column “cycle” denotes the running sequence from 1 to the cycle failure happened; columns {st1, st2, st3} are the operation settings of engines: columns “s1” to “s21” denotes the measurements of sensor 1 to sensor 21. The test dataset also has same column structure of training dataset.

<sup>2</sup> 【Prognostics Center - Data Repository】 | NASA  
<https://ti.arc.nasa.gov/tech/dash/groups/pcoc/prognostic-data-repository/>

## 2.2 Exploratory Data Analysis & Feature Engineering

Before we started to build the models for this case, we did some simple exploratory data analysis (EDA) on training dataset and test dataset. Both them have no missing values. Figure 1 show selected examples of running records from engine 1 to engine 10 on sensor 8 to sensor 11 in scatter plots. We could observe the variation of engines on each sensor. We could observe that there was almost no variation in records of some sensors. Table 3 showed the statistics summary of running cycles of engines in training dataset and test dataset. Generally, we could see that every statistic in training dataset are greater than test dataset. Both the above results may influence the results of our models. So, let's implement simple feature engineering on the dataset.

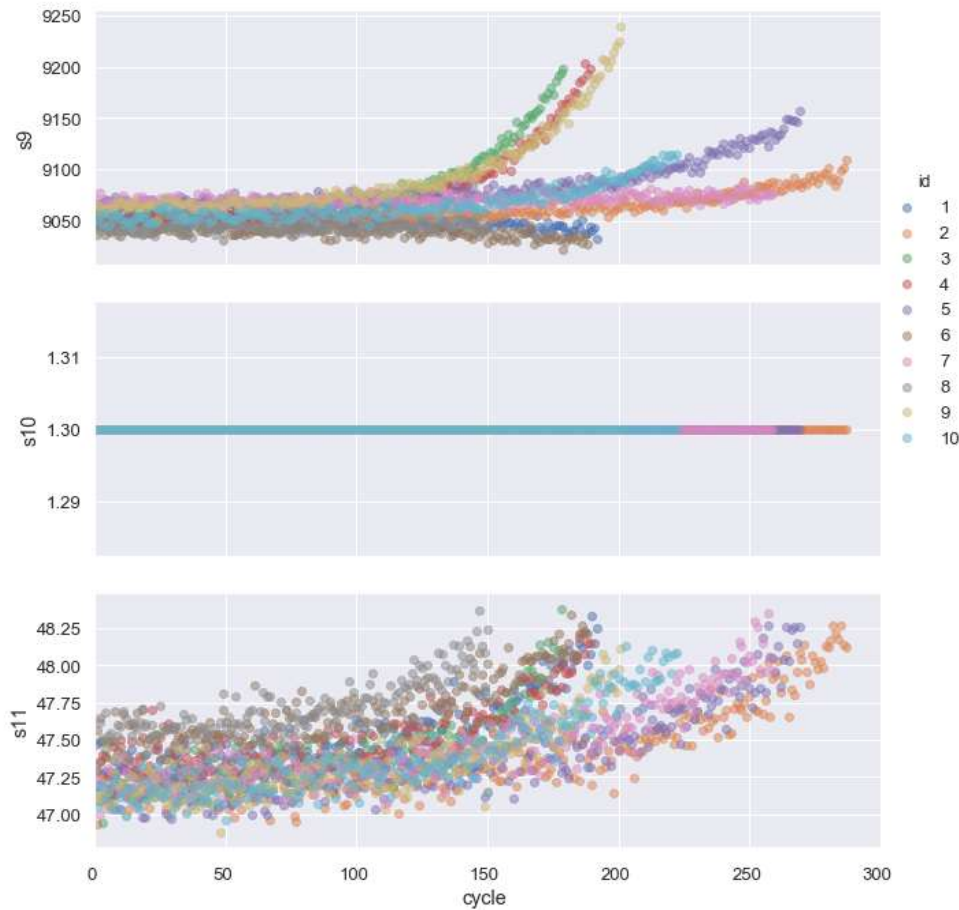


Figure 1 Running records from engine 1 to engine 10 on sensor {9, 10, 11} in scatter plots.

Table 3 Statistics summary of running cycles of engines in training dataset and test dataset

	mean	std	min	25%	50%	75%	max
<b>Training</b>	206.31	46.34	128	177	199	229.25	362
<b>Test</b>	75.52	41.76	7	32.75	86	112.25	145

### 2.2.1 Dimension Reduction

The original features in our dataset are observations of setting 1 to setting 3 and sensor 1 to sensor 21. But according to the previous results, we known that there was almost no variation in records of some sensors. Such these records may be needless for our modeling. According to Figure 2, Figure 3, and Table 4, we known that the variation in {st3, s1, s2, s5, s6, s8, s10, a13, s16, s18, s19} are very tiny and even close to 0. We would remove these features and named the dataset with remining features as “reduced dataset” for experimentation of modeling.

Figure 2 Features' standard deviation of training dataset

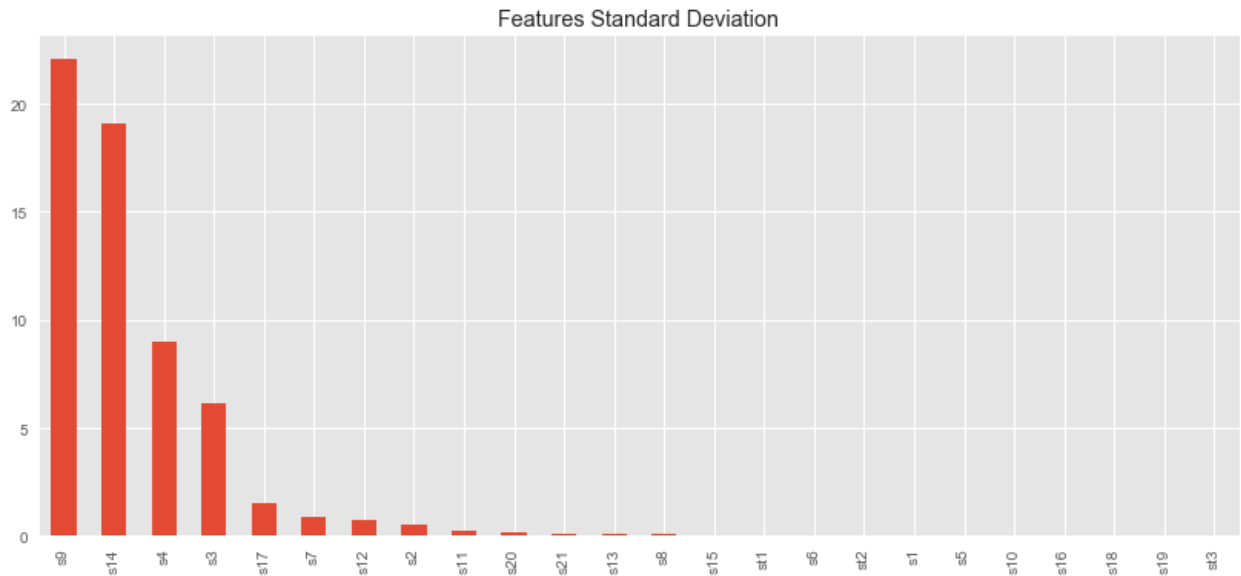
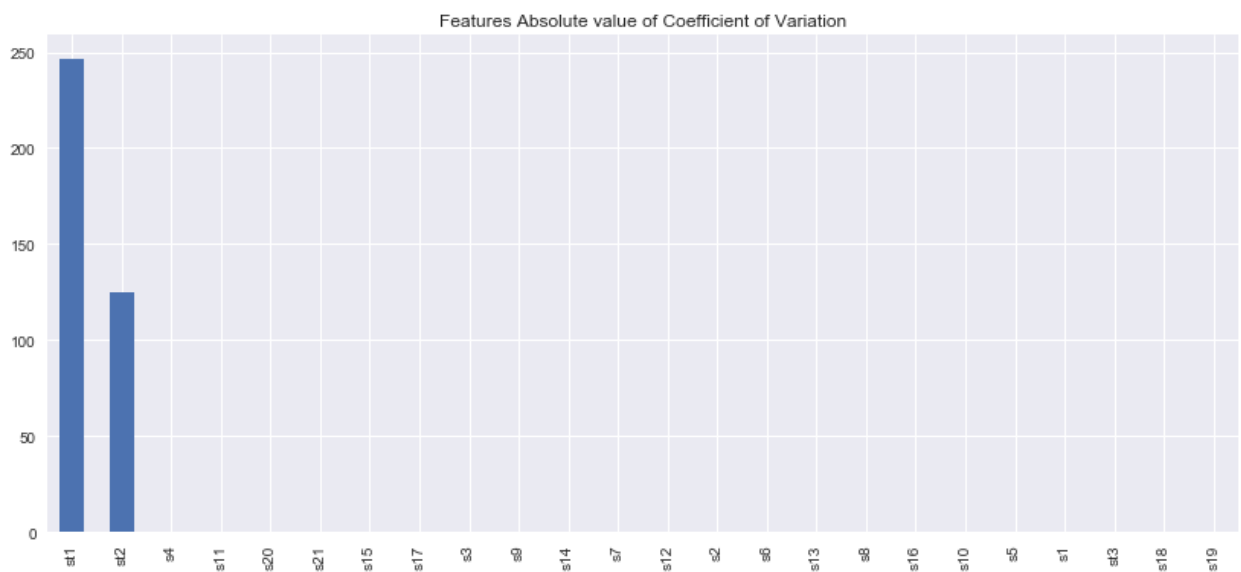


Figure 3 Features' absolute value of coefficient of variation of training dataset



Predictive Maintenance Practice  
A Case of NASA's Aircraft Engines with Data Science Approach

Table 4 Ranking list of features' standard deviation (std) &  
absolute value (abs) of coefficient of variation (c.v.) of training dataset

Variable	std	Variable	abs(c.v.)
s9	22.08288	st1	246.59270
s14	19.07618	st2	124.66320
s4	9.00061	s4	0.00639
s3	6.13115	s11	0.00562
s17	1.54876	s20	0.00466
s7	0.88509	s21	0.00465
s12	0.73755	s15	0.00444
s2	0.50005	s17	0.00394
s11	0.26709	s3	0.00385
s20	0.18075	s9	0.00244
s21	0.10825	s14	0.00234
s13	0.07192	s7	0.00160
s8	0.07099	s12	0.00141
s15	0.03751	s2	0.00078
st1	0.00219	s6	0.00006
s6	0.00139	s13	0.00003
st2	0.00029	s8	0.00003
s1	0	s16	0
s5	0	s10	0
s10	0	s5	0
s16	0	s1	0
s18	0	st3	0
s19	0	s18	0
st3	0	s19	0

### 2.2.2 Feature Construction

Contrast to dimension reduction, we would add some artificial variables from the original data to explore better models. According to Table 3, to prevent from overfitting our models, we decided to add [moving average](#) and [rolling standard deviation](#) with last 10 running cycles of each setting and sensor. Of course, we could use other number for the calculation of moving average and rolling standard deviation to explore potential better models. But in this case, we would just use 10 for an initial test of feature construction. Finally, we named the dataset with plus features as “plus dataset” for experimentation of modeling.

## 3 Regression Modeling

### 3.1 Algorithms & Metrics

In this chapter, we would predict the unexploited remaining useful life of linear and non-linear regression methods. The used algorithms were arranged in Table 5; the performance metrics for the evaluation of regression models are arranged in Table 6.

Table 5 Description of regression algorithms which used in this case

Algorithms	Description <sup>3</sup>
LASSO Regression	LASSO (Least Absolute Shrinkage and Selection Operator) is a type of linear regression method that performs both variable selection and regularization to enhance the prediction accuracy and interpretability. It uses L1 regularization. L1 adds a penalty equal to the absolute value of the magnitude of coefficients.
Ridge Regression	Ridge regression is a way to create a parsimonious model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multicollinearity. It uses L2 regularization. L2 regularization adds an L2 penalty, which equals the square of the magnitude of coefficients.
Polynomial Regression	Polynomial regression is a multiple regression method in which the relationship between the independent variable $x$ and the dependent variable $y$ is modelled as an $n^{\text{th}}$ degree polynomial in $x$ .
Decision Tree Regression	Decision tree learns a hierarchical structure by repeatedly splitting the dataset into separate branches that maximize the information gain of each split. If the target variable can take continuous values (typically real numbers) are called regression trees; if the target variable can take a discrete set of values are called classification trees.
Random Forest Regression	Random forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
XGBoost Regression	XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework and provides a parallel tree boosting that solve many data science problems in a fast and accurate way.

<sup>3</sup> The descriptions of LASSO and Ridge regression were referred from 【Statistics How To: Elementary Statistics for the rest of us!】

<https://www.statisticshowto.datasciencecentral.com/>;

The descriptions of XGBoost was referred from 【xgboost 1.0.0-SNAPSHOT documentation】 <https://xgboost.readthedocs.io/en/latest/>;

The other descriptions of algorithms were referred from 【Data mining - Wikipedia】 [https://en.wikipedia.org/wiki/Data\\_mining](https://en.wikipedia.org/wiki/Data_mining)



Neural Network Regression	Artificial neural networks (ANN) is computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. It can be used in both prediction and classification tasks.
---------------------------	---

Table 6 The performance metrics for the evaluation of regression models

Metrics	Description
RMSE	Root Mean Squared Error of test dataset, where lower value is better.
MAE	Mean Absolute Error of test dataset, where lower value is better.
$R^2$	$R^2$ of test dataset, where higher value is better.
EV	Explained Variance of test dataset, where higher value is better.
mean of Residuals	Mean of residuals between predictions and actual value, where close to 0 may be better. It indicates that should the predictions are too optimistic or not?
$R^2(\text{Train})$	$R^2$ of training dataset, where higher value may be better. Just for reference.

### 3.2 Analyses of Results

The prediction results of all regression models were arranged in Table 7. We used the suffix {O, R, P} to denote the results of model with **O**riginal dataset, **R**educed dataset, and **P**lus dataset, which were derived from feature engineering in the previous chapter. There were total 21 experimental models in this phrase. Rough tuning of hyperparameters were done manually on each model without cross validation. The brief analyses were listed as below:

1. XGBoost has the best regression performance on metrics of {RMSE, MAE,  $R^2$ , EV}, Neural Network and Random Forest rank 2<sup>nd</sup> and 3<sup>rd</sup> in these metrics.
2. Polynomial Regression, Decision Tree have better performance on  $R^2(\text{Train})$ , but they did not perform good scores on the metrics of test dataset.
3. Neural Network has the best performance on mean of residuals, which means the whole distribution of predictions from models may be not so distant from true records.
4. According to metrics of {RMSE, MAE,  $R^2$ , EV}, linear regression model like LASSO and Ridge Regression do not perform well on this case, but they are not the worst model.
5. Generally, Results derived from original features perform better than results derived from reduced features and plus features except from XGBoost and Neural Network. This may be resulted from non-optimal tuning of hyperparameters.
6. The hyperparameter tuning of these results were just done manually. Grid search or random search with cross validation could be helpful for better processing.

Predictive Maintenance Practice  
A Case of NASA's Aircraft Engines with Data Science Approach

Table 7 Metrics scores of results from regression models

Model	RMSE	MAE	R <sup>2</sup>	EV	mean(Residuals)	R <sup>2</sup> (Train)
LASSO (O)	31.974	25.551	0.408	0.667	-21.165	0.580
LASSO (R)	32.449	25.818	0.390	0.654	-21.340	0.578
LASSO (P)	33.798	27.259	0.339	0.640	-22.835	0.597
Ridge (O)	31.955	25.541	0.409	0.668	-21.171	0.580
Ridge (R)	32.392	25.765	0.392	0.654	-21.234	0.578
Ridge (P)	33.524	27.033	0.349	0.646	-22.645	0.593
Polynomial (O)	31.775	24.188	0.415	0.640	-19.697	<b>0.627</b>
Polynomial (R)	32.202	24.455	0.400	0.621	-19.561	<b>0.622</b>
Polynomial (P)	34.800	26.290	0.299	0.505	-18.852	<b>0.704</b>
Decision Tree (O)	32.075	24.138	0.404	0.635	-19.971	<b>0.643</b>
Decision Tree (R)	32.482	24.519	0.389	0.622	-20.065	<b>0.624</b>
Decision Tree (P)	33.980	25.031	0.331	0.578	-20.625	<b>0.680</b>
Random Forest (O)	27.849	21.697	0.551	0.750	-18.558	0.617
Random Forest (R)	28.742	21.892	0.522	0.726	-18.776	0.634
Random Forest (P)	29.379	24.116	0.500	0.754	-20.927	0.620
XGBoost (O)	<b>24.544</b>	<b>19.237</b>	<b>0.651</b>	<b>0.782</b>	15.056	0.243
XGBoost (R)	<b>24.783</b>	<b>19.474</b>	<b>0.644</b>	<b>0.774</b>	14.936	0.240
XGBoost (P)	<b>23.951</b>	<b>18.384</b>	<b>0.668</b>	<b>0.798</b>	14.979	0.364
Neural Network (O)	25.444	21.558	0.625	0.633	-3.789	0.316
Neural Network (R)	28.859	23.775	0.518	0.518	-0.172	0.255
Neural Network (P)	26.150	21.427	0.604	0.614	-4.167	0.413

Though XGBoost regression got the best performance on metrics of {RMSE, MAE, R<sup>2</sup>, EV}, but let's see Figure 4 and Figure 5. We could observe that most predictions are greater than actual values in paired data. Besides, only mean of residuals in set of XGBoost regression models are positive. This indicate that the results of XGBoost regression may be too optimistic in issue of predictive maintenance. If we use the results by XGBoost regression for advanced modeling, we may get unfavorable outcomes in real environment.

By contrast, results of Neural Network seem not so optimistic as XGBoost regression according to mean of residuals. But let's review the back to Table 3, we known that there are some obvious differences between whole training dataset and test dataset. On this perspective, Neural Network regression may be still overfitted from our training dataset. For a conservative perspective, the results of Random Forest regression may be the most acceptable in our experimentation of regression modeling. It is right the reason that why we filled a special color in rows of Random Forest in Table 7.

# Predictive Maintenance Practice A Case of NASA's Aircraft Engines with Data Science Approach

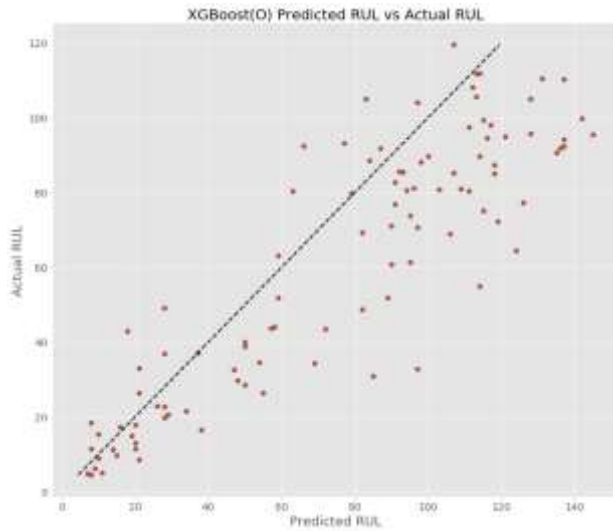


Figure 4 Predicted RUL vs Actual RUL in XGBoost (O)<sup>4</sup>

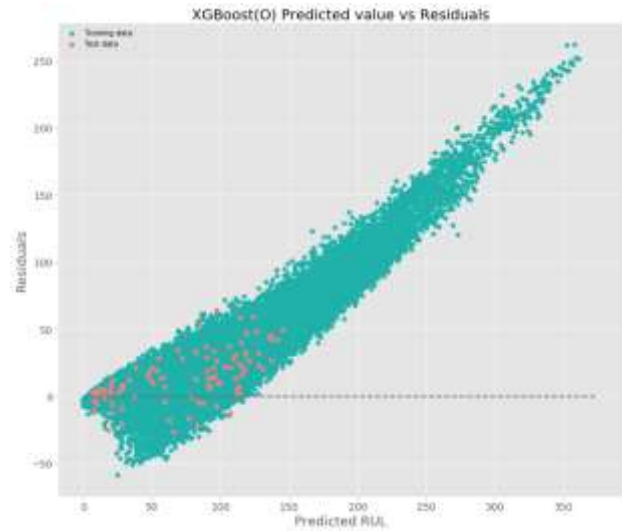


Figure 5 Predicted value vs Residuals in XGBoost (O)

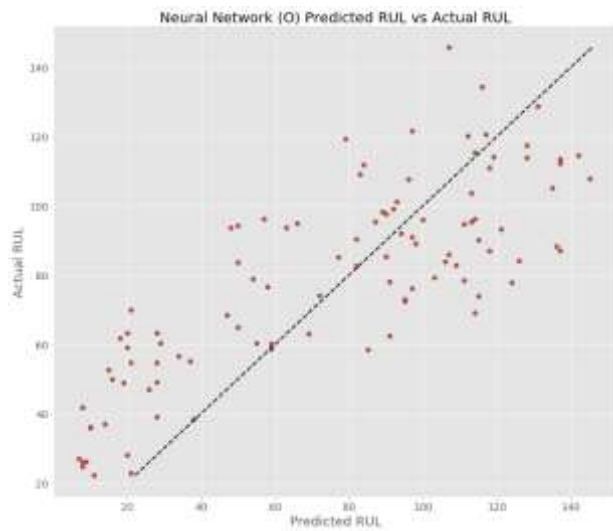


Figure 6 Predicted RUL vs Actual RUL in Neural Network (O)

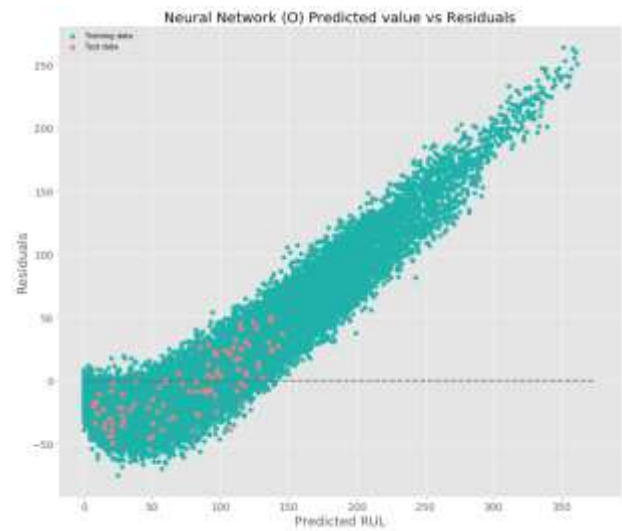


Figure 7 Predicted value vs Residuals in Neural Network (O)

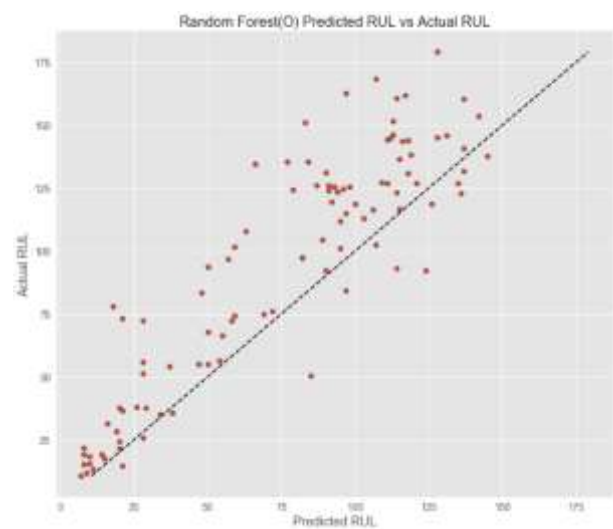


Figure 8 Predicted RUL vs Actual RUL in Random Forest (O)

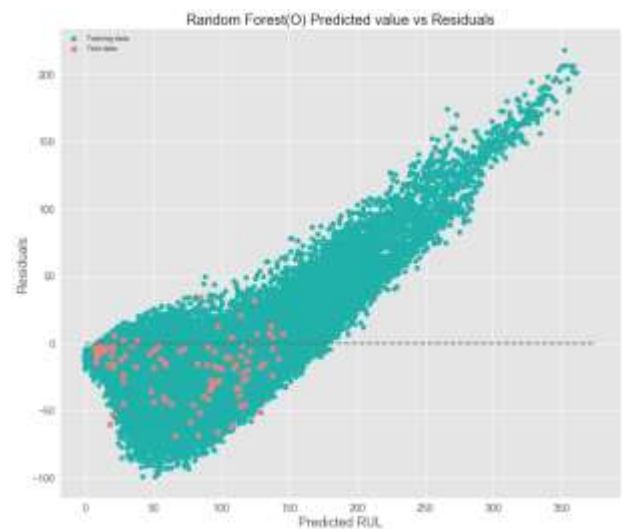


Figure 9 Predicted value vs Residuals in Random Forest (O)

<sup>4</sup> More figures of results were deposited on my GitHub: <https://github.com/littleHurt/Predictive-Maintenance-Practice/tree/master/Figures>

## 4 Classification Modeling

### 4.1 Algorithms & Metrics

In this chapter, we would build classification models based on the results of predictive models from previous chapter.<sup>5</sup> The used algorithms were arranged in Table 8. Besides, **Decision Tree Classification**, **Random Forest Classification**, **XGBoost Classification**, and **Neural Network Classification** were also used for classification modeling, and relevant descriptions of these algorithms were already arranged in Table 5. The performance metrics for the evaluation of classification models are arranged in Table 10, we used elements of confusion matrix such like Table 9 to support Table 10.

Table 8 Description of classification algorithms which used in this case

Algorithms	Description <sup>3</sup>
Logistics Regression	Logistic regression is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.
$k$ -NN Classification	$k$ -nearest neighbors algorithm is a non-parametric method. The input consists of the $k$ closest training examples in the feature space. The output depends on whether $k$ -NN is used for classification or regression. It is a type of lazy learning, where the function is only approximated locally and all computation is deferred until classification.
SVM Classification	Support Vector Machines (SVM) model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. It can efficiently perform a non-linear classification by the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.
Gaussian Naive Bayesian Classification	Naive Bayesian classifiers are a family of simple probabilistic classifiers based on Bayesian theorem with strong independence assumptions between the features. And Gaussian Naive Bayesian classifier is used to deal with continuous data with assumption that the data associated with each class are distributed according to a normal distribution.

Table 9 A simple example of confusion matrix<sup>6</sup>

Total population	<i>True Condition positive</i>	<i>True Condition negative</i>
<i>Predicted condition positive</i>	True positive (TP)	False positive (FP)
<i>Predicted condition negative</i>	False negative (FN)	True negative (TN)

<sup>5</sup> We used the results of regression models as base of labeling for classification modeling.

<sup>6</sup> 【Confusion matrix – Wikipedia】 [https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix)

Table 10 Metrics scores of results from classification models

Metrics	Descriptions <sup>6</sup>
Accuracy	$= (TP + TN) / (TP + TN + FP + FN)$
Precision	$= TP / (TP + FP)$ , where is a trade-off value to Recall.
Recall	$= TP / (TP + FN)$ , where is a trade-off value to Precision.
F1 Score	$= 2TP / (2TP + FP + FN)$ , where is a balanced metrics between Precision and Recall.
ROC-AUC	Area Under Curve (AUC) of Receiver Operating Characteristic (ROC). It is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings, where $TPR = TP / (TP + FN)$ and $FPR = FP / (FP + TN)$ .

## 4.2 Analyses of Results

All metrics in Table 10 are ranged from 0 to 1, where higher value is better. The metrics scores of results from classification models were arranged in Table 11. We used the suffix {O, R, P} to denote the results of model with **O**riginal dataset, **R**educed dataset, and **P**lus dataset, which were derived from feature engineering in the previous chapter. There were total 24 experimental models in this phrase. Some tuning of hyperparameters were implemented automatically by a grid search function on each model with 6 folds cross validations to explore better models. The brief analyses were listed as below:

1. Most models perform well on Accuracy, there are about 22 models got 90 or near 90 on Accuracy.
2. Lot of models perform well on Precision, there are about 18 models got 90 or near 90 on Precision.
3. Most models perform well on ROC-AUC, there are about 23 modes got 90 or near 90 on ROC-AUC score.
4. Unfortunately, few models perform well on Recall, only 4 models got 90 or near 90 on Recall, but Recall should be deemed the most important metrics in the issue of predictive maintenance.
5. In our experiments, Gaussian Naive Bayesian models almost got best performance with high scores on all metrics. We also shown its relevant metrics plots on Figure 10 to Figure 13.
6. Generally, Logistics regression,  $k$ -NN, SVM classification models with original features perform slightly better than same models of algorithm with reduced features.
7. For these 8 kinds algorithm of classification models, almost over half of them (Logistics regression,  $k$ -NN, SVM, Decision Tree, XGBoost classification) perform better with plus features.
8. For these 8 kinds algorithm of classification models, Random Forest is the most stable model which the results with three kinds of features are almost same.

Predictive Maintenance Practice  
A Case of NASA's Aircraft Engines with Data Science Approach

Table 11 Metrics scores of results from classification models

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression (O)	0.88	0.933	0.56	0.700	0.977
Logistic Regression (R)	0.88	0.933	0.56	0.700	0.979
Logistic Regression (P)	0.90	0.941	0.64	0.762	0.979
k-NN (O)	0.90	0.941	0.64	0.762	0.894
k-NN (R)	0.90	0.941	0.64	0.762	0.936
k-NN (P)	0.91	0.944	0.68	0.791	0.960
SVM (O)	0.91	0.944	0.68	0.791	0.892
SVM (R)	0.90	0.941	0.64	0.762	0.905
SVM (P)	0.92	0.947	0.72	0.818	0.891
Gaussian Naive Bayesian (O)	0.94	0.828	<b>0.96</b>	0.889	0.988
Gaussian Naive Bayesian (R)	0.93	0.846	<b>0.88</b>	0.863	0.988
Gaussian Naive Bayesian (P)	0.94	0.852	<b>0.92</b>	0.885	0.983
Decision Tree (O)	0.91	0.900	0.72	0.800	0.971
Decision Tree (R)	0.90	0.857	0.72	0.783	0.949
Decision Tree (P)	0.93	0.950	0.76	0.844	0.929
Random Forest (O)	0.92	0.947	0.72	0.818	0.979
Random Forest (R)	0.92	0.947	0.72	0.818	0.984
Random Forest (P)	0.92	0.947	0.72	0.818	0.978
XGBoost (O)	0.92	0.947	0.72	0.818	0.981
XGBoost (R)	0.91	0.944	0.68	0.791	0.984
XGBoost (P)	0.91	0.944	0.68	0.791	0.993
Neural Network (O)	0.80	0.857	0.24	0.375	0.808
Neural Network (R)	0.67	0.426	<b>0.92</b>	0.582	0.889
Neural Network (P)	0.89	0.938	0.60	0.732	0.971

We wanted to select some models with top performance for composite comparison. The metrics priority we concerned is: *Recall*  $\rightarrow$  *F1 Score*  $\rightarrow$  *ROC-AUC*  $\rightarrow$  *Precision*.

Gaussian Naive Bayesian models with all kinds of features seem good candidates, but we decided remove the model with reduced features which got worst result in models of Gaussian Naive Bayesian; Neural Network model with reduced features seem a good candidate which also perform well on Recall (0.92), but its Precision is too low (0.426). So, we would not consider it. Follow by top of ranking list of point 1, we would select models below for advanced comparison and the plotted results were shown in Figure 14 and Figure 15.

Gaussian Naive Bayesian (O) & (P), Decision Tree (P), Random Forest (R), XGBoost (O)

# Predictive Maintenance Practice A Case of NASA's Aircraft Engines with Data Science Approach

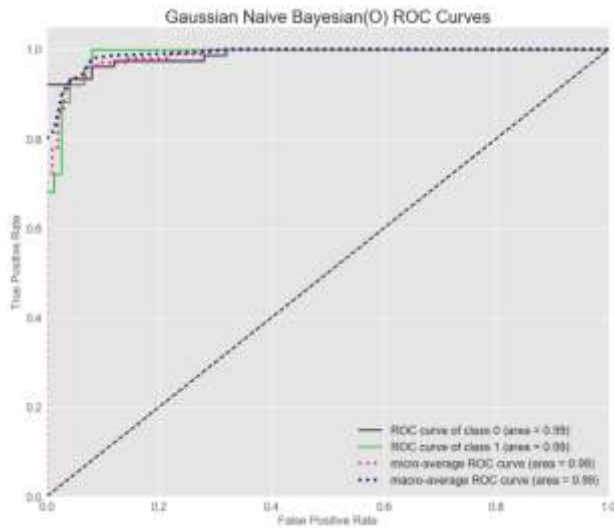


Figure 10 ROC curves of Gaussian Naive Bayesian (O)<sup>7</sup>

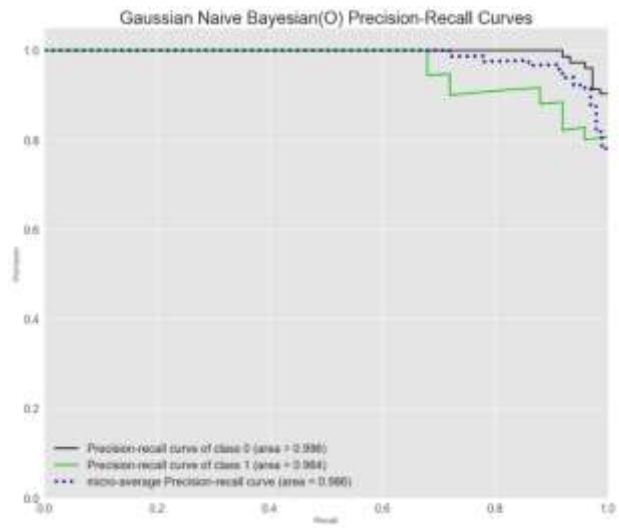


Figure 11 Precision-Recall curves of Gaussian Naive Bayesian (O)

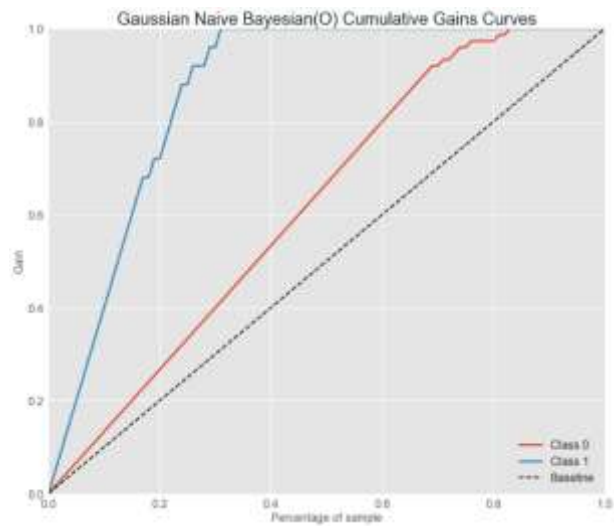


Figure 12 Cumulative Gains Curves of Gaussian Naive Bayesian (O)

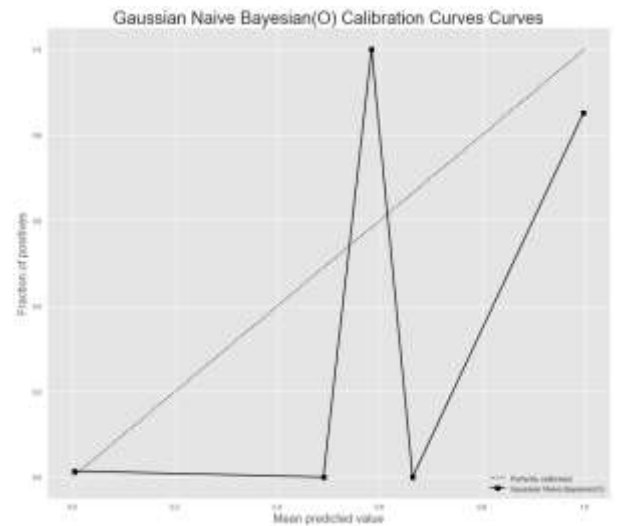


Figure 13 Calibration Plots of Gaussian Naive Bayesian (O)

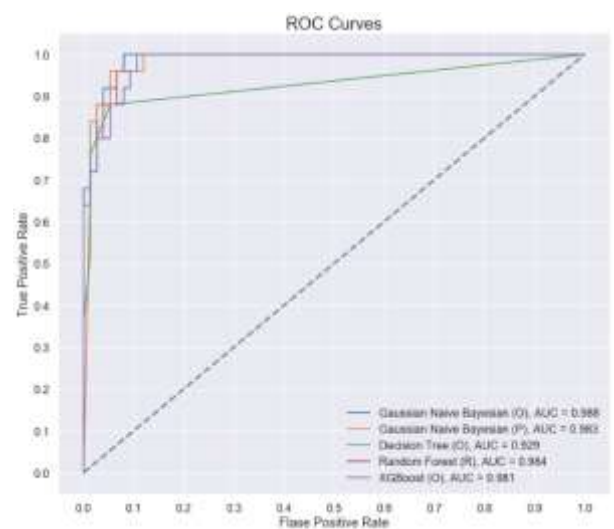


Figure 14 ROC curves of selected top models

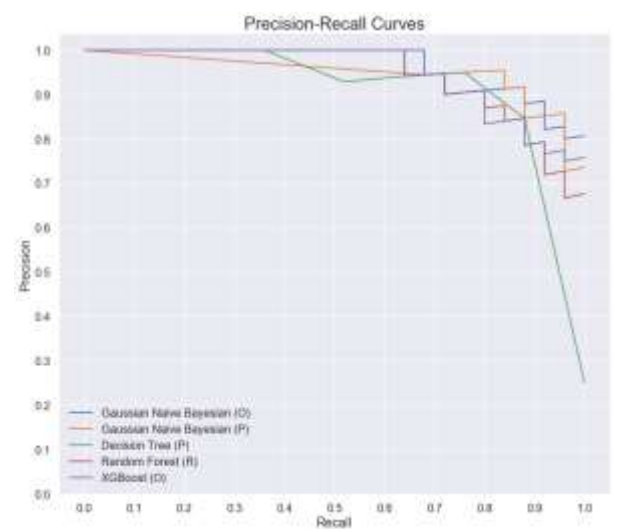


Figure 15 Precision-Recall curves of curves of selected top models

<sup>7</sup> More figures of results were deposited on my GitHub: <https://github.com/littleHurt/Predictive-Maintenance-Practice/tree/master/Figures>



### 4.3 Benefits Evaluation

Apart from traditional metrics, according to Susto, G. A. etc. (2015)<sup>8</sup>, we could calculate benefit of predictive maintenance models from a customized cost-based formula. The similar idea was also referred by Foster Provost and Tom Fawcett (2013)<sup>9</sup>. Let's define *TEC* as "Total Expected Cost" derived from predictive maintenance modules. The descriptions of parameters of *TEC* were arranged in Table 12 and it was supported by Table 9. The results of calculation of benefits of models were arranged in Table 13, where we used the models in Figure 14 and Figure 15 as examples.

$$TEC = mUL * pUL * cUL + pUB * cUB$$

Table 12 the descriptions of parameters of *TEC*

Parameters	Descriptions
<b><i>mUL</i></b>	Amount of Unexpected remaining useful Lifetime, it should be assumed from results of regression models. We assumed it with 25 cycles for example in this case.
<b><i>pUL</i></b>	Percentage of over predicted safe samples, where $pUL = FP / (TP + TN + FP + FN)$
<b><i>pUB</i></b>	Frequency of Unexpected Break, where $pUB = FN / (TP + TN + FP + FN)$
<b><i>cUB</i></b>	The relevant cost to <i>pUB</i> . We assigned it with \$USD 5,000 for example in this case.
<b><i>cUL</i></b>	The relevant cost to <i>pUC</i> . We assigned it with 200 times cost of <i>cUB</i> in this case.

Table 13 The results of calculation of benefits of PdM models

Model	Cost	TP	FP	FN	TN
Gaussian Naive Bayesian (O)	\$ 16,250	70	5	1	24
Gaussian Naive Bayesian (P)	\$ 25,000	71	4	2	23
Decision Tree (P)	\$ 61,250	74	1	6	19
Random Forest (R)	\$ 71,250	74	1	7	18
XGBoost (O)	\$ 71,250	74	1	7	18

According to Table 13, Gaussian Naive Bayesian with original features got minimal cost from the customized formula. The relevant parameters of cost were worthy to consults with professional aerospace business for a much accurate evaluation.

<sup>8</sup> 《Machine Learning for Predictive Maintenance - A Multiple Classifier Approach》 Susto, G. A. etc. (2015)

<sup>9</sup> 《Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking》 Foster Provost and Tom Fawcett. (2013)



## 5 Conclusions

### 5.1 Summary

We have built a set of predictive maintenance modules by data science approach on case of maintenance of aircraft engines and used a customized function to calculate the expected cost to selected the best module. Our results could answer of two key questions:

1. How many unexploited running cycles do the engines remain?
2. Whether the engines will break in the following cycle?

Besides, some domain knowledges were also applied in the project to fit the reality.

### 5.2 Extension of Future

This case is just a simple off-line practice. The following actions may be helpful for extension:

1. Consult to domain experts of aircraft mechanical engineers
2. Try more feature engineering and experiments cross-validation
3. More detailed hyperparameters tuning for machine learning models.
4. Try to build ensemble models by combining with multiple classification threshold.

## References

The sequences of references were list by the importance to this project.

1. **【predictive-maintenance | GitHub】** by Samimust  
<https://github.com/Samimust/predictive-maintenance>
2. **【Predictive Maintenance ML (IIoT) | Kaggle】** by Bill Stuart  
<https://www.kaggle.com/billstuart/predictive-maintenance-ml-iiot>
3. **【Prognostics Center - Data Repository | NASA】**  
<https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>
4. **【scikit-learn: machine learning in Python — scikit-learn 0.21.3 documentation】**  
<https://scikit-learn.org/stable/index.html>
5. **【Drawing multiple ROC-Curves in a single plot · Abdullah Al Imran】**  
<https://abdalimran.github.io/2019-06-01/Drawing-multiple-ROC-Curves-in-a-single-plot>
6. **【Metrics Module (API Reference)】** Scikit-plot documentation  
<https://scikit-plot.readthedocs.io/en/stable/metrics.html>
7. **【xgboost 1.0.0-SNAPSHOT documentation】**  
<https://xgboost.readthedocs.io/en/latest/>
8. **【Data mining - Wikipedia】**  
[https://en.wikipedia.org/wiki/Data\\_mining](https://en.wikipedia.org/wiki/Data_mining)
9. **【Confusion matrix – Wikipedia】**  
[https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix)
10. **【Statistics How To: Elementary Statistics for the rest of us!】**  
<https://www.statisticshowto.datasciencecentral.com/>
11. 《*Machine Learning for Predictive Maintenance - A Multiple Classifier Approach*》 IEEE Transactions on Industrial Informatics, 11(3), 812-820. (2015); Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A.  
<https://pureadmin.qub.ac.uk/ws/portalfiles/portal/17844756/machine.pdf>
12. 《*Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*》 1<sup>st</sup> edition (2015); O'Reilly Media; Foster Provost & Tom Fawcett  
<https://www.amazon.com/Data-Science-Business-Data-Analytic-Thinking/dp/1449361323>
13. 〈 Predictive Maintenance: What is PdM Maintenance? 〉 | Fiix  
<https://www.fiixsoftware.com/maintenance-strategies/predictive-maintenance/>