Comparison of binary classifiers for data-driven prognosis of jet engines health

Jean-Loup Loyer[1], Elsa Henriques[2], and Steve Wiseall[3]

[1,2]*Instituto Superior Técnico, Universidade de Lisboa, Lisbon,1049-001,Portugal*
jean-loup.loyer@tecnico.ulisboa.pt
elsa.henriques@tecnico.ulisboa.pt

[3]*Rolls-Royce plc, Derby, DE24 8BJ, United Kingdom*
steve.wiseall@rolls-royce.com

ABSTRACT

A reliable prognosis is crucial to manage asset health and predict maintenance needs of large civil jet engines, which in turn contribute to enhanced aircraft airworthiness, longer time on wing and optimized lifecycle costs. With the accumulation of large amount of data over the last decade, one can relate the number of components serviced during a maintenance visit to the history of parameters inside and outside the engine (temperatures, pressure, shaft rotation speeds, vibration levels, etc.). While established statistical models had been developed for small samples, more recent computer-intensive statistical techniques from the field of Machine Learning (ML) can handle more complex datasets. In particular, binary classifiers constitute an attractive option to predict the probability of servicing the components of a given jet engine at the next maintenance visit. This paper demonstrates the validity of such data-driven methods on an industrial case study involving failures of thousands of compressor blades in aeronautical turbomachines. The prediction accuracy obtained with the ML techniques presents a significant improvement over the state-of-the-art. Moreover, the performance of six binary classifiers with different characteristics - logistic regression, support vector machines, classification trees, random forests, gradient boosted trees and neural networks - was compared according to four qualitative and quantitative criteria. Results show that there is no clear winner, although ensemble models based on trees (random forests and boosted trees) offer a good overall compromise while neural networks offer the best absolute performance. In the industrial world, the business objectives, the environment in which the models are deployed and the users' skills should dictate the choice of the most adequate statistical technique.

1. INTRODUCTION AND MODELING APPROACH

A jet engine is complex machine subject to particularly demanding operating conditions that explains the deterioration of the engine components. Therefore, a proper maintenance of jet engines is crucial to ensure airworthiness, reduce fuel consumption and ultimately lower operating cost. Large amount of data are acquired on a permanent basis by jet engine manufacturers to help engineers in predicting future maintenance needs. On the one hand, the health of a jet engine is monitored in real-time by dozens of sensors measuring hundreds of variables inside and outside the engine (temperature, pressure, rotation speeds, vibration levels…); the data from this Engine Health Monitoring (EHM) system are recorded in corporate databases for later analysis. On the other hand, for every maintenance shop visit performed all around the world, the number of components scrapped are registered by maintenance technicians and engineers and sent to the engine manufacturer. These two major types of data can be combined to establish a prognosis of the health of the fleet of engines.

To predict future maintenance needs, reliability engineers rely on multiple techniques that can be divided into two categories:

1. Analytical inductive methods based on engineering reasoning and analysis of failure modes of the part, such as Failure mode, effects, and criticality analysis (FMECA) (Jordan, 1972). Methods in this category typically require a deep technical expertise and knowledge of the product but limited volumes of historical data.
2. Deductive statistical techniques inferring risk of failure using actual past data from similar cases. Many such statistical methods have been applied to reliability engineering (Meeker & Escobar, 1998): analysis of

failure time data (Kalbfleisch & Prentice, 2011), biostatistics-inspired survival analysis (Lawless, 2003), Poisson-related models based on count data with excess zeros such as zero-inflated models (Lambert, 1992) or hurdle models (Grogger & Carson, 1991).

This paper covers a set of methods belonging to the second category. Most of such statistical methods currently in use by industrial corporations are based on established statistical models developed for dealing with small samples. However, the large volumes of monitoring data acquired over the last decade allowed resorting to more data-driven, computer-intensive methods for predicting maintenance needs (Jardine, Lin, & Banjevic, 2006). In this paper, we intend to validate a statistical approach for PHM of jet engine parts based on binary classifiers from the field of machine learning. Such binary classifiers predict whether a given part in the engine is likely to be scrapped (output variable $Y = 1$) or not ($Y = 0$) at the next maintenance visit, given its own history and the history of similar components. Compared to the aforementioned statistical models, the literature on binary classifiers applied to reliability engineering and maintenance planning is still scarce. For instance, Kim, Tan, Mathew, Kim, and Choi (2008) used Support Vector Machines to predict the remaining useful life of elements in high pressure liquefied natural gas pumps. Caesarendra, Widodo, and Yang (2010) used logistic regression to evaluate the degradation of bearings. Rafiee, Arvani, Harifi, and Sadeghi (2007) used neural networks to monitor the condition of gearbox components. Nevertheless, there are few examples in the literature comparing rigorously the predictive performance of several binary classifiers concurrently, which is the objective of our paper.

According to our approach, the statistical model of part failure can be expressed in the most general way as:

$$Y = f_\theta(X) + \varepsilon \tag{1}$$

where Y is the $N \times 1$ output vector to be predicted (containing the probability of failure or the occurrence of failure in our case study), X is the $N \times (p+1)$ matrix of predictors (including intercept) , f_θ is the actual function to estimate and ε is an $N \times 1$ vector of residuals (i.e. errors) of the model. The function f_θ is potentially complex, nonlinear and depends on a set of parameters θ – varying from model to model - to be estimated via model fitting. In equation (1), the residuals ε are assumed to be centered and normally distributed with constant variance σ^2, i.e. $\varepsilon \sim \mathcal{N}(0, \sigma^2)$). In fact, it is impossible to identify the actual function f_θ: instead, the statistical models provide an estimate $\hat{f}_\theta$ of the actual f_θ. The role of the statistician is to find the model that is as close as possible to f_θ by 1) finding the most relevant type of model and 2) by tuning the parameters θ.
Following the description of the context in this introduction, the article will present the case study and the dataset in Section 2 before detailing the methodology – including simple mathematical formulation behind the classifiers - in Section 3. The results of the comparison of the binary classifiers are covered in Section 4 and commented in Section 5, which also opens discussion for potential improvements and next steps.

2. CASE STUDY AND DATASET DESCRIPTION

2.1. Description of the compressor blades

We tested the validity of our approach on a case study involving blades in the intermediate pressure (IPC) and high pressure compressors (HPC) of Rolls-Royce Trent 500 engines (Figure 1). Such compressor blades are made of titanium (in the front and middle stages) or nickel alloys (in the rear stages of the HPC) and manufactured by forging or machining processes. We selected compressor blades as our case study as they are relevant candidates to test the validity of the statistical approach:

- There are numerous stages in a Trent 500 (8 in the IPC and 6 in the HPC), each containing dozens of blades. Thus, in total, there are hundreds of compressor blades in a Trent 500 engine. This leads to a dataset with more observations (higher N), an important condition for making the statistical approach viable.
- The 14 different types of blades are located all along the gas path of the engine, and therefore subject to very diverse operating conditions (amplitude of temperature and pressure, rotation speeds and vibration levels in the IPC and HPC shafts, etc.). This large diversity of situations also improves the meaningfulness and quality of the statistical estimates.

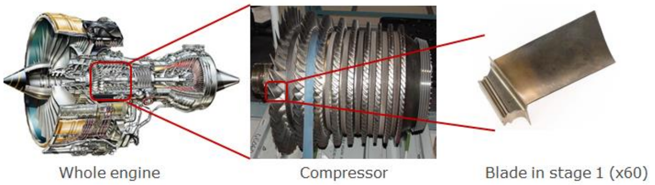

Figure 1 - Location of the blades in Trent 500 engine

Knowing the deterioration mechanisms of the component is important to select the most adequate type of statistical model and the predictors entering the model as inputs. Compressor blades are subject to demanding conditions during engine operations and their degradation is influenced mostly by gas temperature and pressure, shaft rotation speed and vibration levels.

2.2. Structure of the dataset

Our dataset comprises a total of $N = 12132$ serviced components, corresponding to 36 components per engine, for 337 maintenance visits performed on 176 different engines between 2000 and 2012. The number of components serviced during the maintenance visits comes from the analysis of maintenance invoices while the predictors of the model have been extracted from the Engine Health Monitoring (EHM) database.

Out of the hundreds of variables recorded by the EHM system, we have selected only the $p = 11$ most relevant ones (Table 1). Selecting a limited number of variables fulfills several objectives: 1) keep the predictors most pertinent to the failure mechanisms of the components in the case study, 2) make the approach more tangible for the reader by exposing few, meaningful variables and 3) not to compromise industrial confidentiality.

Table 1. Variables selected as model predictors.

Variable	Description
Cycles	Number of cycles (i.e. flights) done by the engine between two maintenance visits
Average TGT margin	Average of the margin (i.e. difference with the admissible value) of the temperature in the turbine (highest temperature in the engine) over the period between two maintenance visits
Delta TGT margin	Difference between the initial and final values of the turbine temperature margin over two maintenance visits
Average N2 margin	Average of the margin of the rotation speed of the intermediate shaft over the period between two maintenance visits
Delta N2 margin	Difference between the initial and final values of the intermediate shaft speed margin over two maintenance visits
Average N2	Average of the absolute rotation speed of the intermediate shaft over the period between two maintenance visits
Delta N2	Difference between the initial and final values of the intermediate shaft absolute speed over two maintenance visits
Average VB2	Average of the vibration level in the intermediate shaft over the period between two maintenance visits
Delta VB2	Difference between the initial and final values of the intermediate shaft vibration level over two maintenance visits
Average OAT	Average of the absolute Outside Air Temperature over the period between two maintenance visits
Delta OAT	Difference between the initial and final values of the absolute Outside Air Temperature over two maintenance visits

The number of cycles and the Turbine Gas Temperature (TGT) are usually considered by maintenance engineers as the best proxies for overall deterioration of a jet engine. The rotation speed N2 of the intermediate shaft can be considered as a proxy for fatigue due to centrifugal forces and fluid-structure interaction. Taking both the margin and absolute values of some of those engine parameters allowed us to include in the statistical models two complementary types of information about the engine operations. The average and delta values over the period between two maintenance visits provide us with information about the average and variability of the engine parameters, respectively.

3. METHODOLOGY

In this Section 3, we describe the general modeling approach that we followed, as well as the simple characteristics of the binary classifiers compared in the paper.

3.1. General approach and data cleaning

Our objective is to obtain an estimate $\hat{f}_\theta$ that is as close as possible to the actual true function f_θ explaining Y as a function of the engine parameters defined in the matrix X of predictors in equation 1. The choice of the model $\hat{f}_\theta$ depends on the probability distributions of the output Y and the structure of the predictors X. The predictors X being all numeric, it is possible to use a large variety of models. After preliminary data exploration, we found that the probability distribution of the output Y discarded models based on the Poisson distribution or count data. Instead, binary classifiers appeared more adapted to our case study: the output Y thus takes the value of 1 for a failed component and 0 for a non-failed component. Y can alternatively be a probability of failure, in which case a threshold has to be defined so as to classify the probability as corresponding – or not - to a failed component.

The list of predictors X being defined, we pre-processed the data to make them more suitable for subsequent statistical analysis. First, the few outliers were removed or their value reattributed by usual imputations techniques: imputations of the mean or median by relevant groups and regression on other predictors. Second, the predictors were scaled[1] in order to increase the quality of the estimates and increase the convergence of the algorithms, as some are known for their instability, notably neural networks. Scaling the predictors thus ensures giving a common ground for comparing all the binary classifiers.

[1] Scaling means that each predictor X_i in X was transformed as $X'_i = (X_i - \mu_{X_i})/\sigma_{X_i}$ where μ_{X_i} and σ_{X_i} are respectively the mean and standard error of the variable X_i

3.2. Characteristics of binary classifiers

Many binary classifiers have emerged from the field of Machine Learning over the last two decades to predict phenomena involving binary outcomes in a large diversity of applications (e.g. disease diagnosis, image recognition). We cover in this section 6 of the most popular ones, presenting a gradual increase in terms of model complexity, from the easily interpretable linear logistic regression to the complex, highly nonlinear neural networks. This section is based on the widely acknowledged text of Hastie, Tibshirani, and Friedman (2009) and intends to initiate the reader – especially the one not versed in statistical methods – to the main characteristics and differences between binary classifiers. An intuitive presentation of the principle and a simple formulation of the mathematics of the techniques are presented; moreover, the tuning parameters (aka hyperparameters) of the models are described so as to illustrate their complexity. Indeed, behind the sometimes fancy names attributed by statisticians or computer scientists to those binary classifiers, one should be aware of the explanatory power and predictive accuracy of these techniques, but also the amount of skills required to use them properly.

3.2.1. Logistic regression

In logistic regression the binary output Y is transformed so that the natural logarithm of its odds[2] is expressed as a linear function of X, the matrix of predictors. It can also be written as the probability[3] of the outcome of Y given X:

$$\begin{aligned} &\log\left(\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right) = \beta_0 + \beta^T x \\ &P(Y=1|X=x) \\ &= \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)} \\ &= 1 - P(Y=0|X=x) \end{aligned} \tag{1}$$

Therefore, the logistic regression is a binary classifier depending on a linear function of the predictors. The model provides the linear coefficients β_i that quantify the risks on the output Y:

$$\exp(\beta_i) = 1 + \alpha \tag{2}$$

where α is the increase (and symmetrically decrease if $1 + \alpha < 1$) of the relative risk of failure provoked by an increase in one unit of the predictor X_i. Thanks to the coefficients of the logistic regression, it is thus possible to estimate the marginal effect of each predictor on the output variable Y, rendering the model more interpretable. This unique characteristic combined with the simplicity of the model assumptions makes the logistic regression particularly attractive and popular amongst analysts with little statistical background. Moreover, the logistic regression doesn't require tuning parameters (aka hyperparameters) which often represent a considerable part of the modeling process.

Nonetheless, the deterioration of jet engine components is a nonlinear stochastic process and predictors are usually correlated. Unable to capture this added complexity of the dataset, the logistic regression is limited in terms of goodness-of-fit and prediction accuracy: more sophisticated models thus have to be used.

3.2.2. Support Vector Machines

Support Vector Machines (SVM) became popular two decades ago after the research on statistical learning theory of Vapnik (1996). It is a nonlinear and non-parametric method based on transforming, via a complex transform function Φ, the initial (often non separable) dataset into a new space of much higher – and potentially infinite – dimension. In this new space, the likelihood of having a separable dataset is much higher and it becomes possible to obtain a linear decision boundary, in lieu of a nonlinear decision boundary in the initial space. In this article, we used a particular type of SVM classifier called "C-SVM" which can be formulated mathematically as an optimization problem under constraints (Chang et al., 2011):

$$\begin{aligned} &\min_{\beta,\beta_0,\xi} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i \ \text{ under} \\ &y_i(\langle\beta,\ \Phi(x_i)\ \rangle + \beta_0) \geq 1 - \xi_i \qquad \text{and} \\ &\xi_i \geq 0, i = 1, \dots, N \end{aligned} \tag{3}$$

In equation (3), y_i is the i^th element of the relabeled[4] version of the output Y, x_i is the i^th element of the initial matrix of predictor X, β is the vector of coefficients, β_0 a constant (i.e. intercept) and $\xi_{i\in\{1,\dots,N\}}$ are parameters quantifying the degree of non separability of the elements in the dataset. Geometrically speaking, solving this problem consists in determining the hyperplane such as the estimated values $\langle\beta,\ \Phi(x_i)\ \rangle + \beta_0$ don't deviate from the output values y_i. The aforementioned mapping from the initial low dimensional space to a higher dimensional space is done by so-called kernel functions $k(x_i, x_j)$ expressed as the inner product of the transform function Φ i.e. $k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. There are several types of kernel functions: linear $x_i \cdot x_j$, d-degree polynomial $(\gamma x_i \cdot x_j + C)^d$, Gaussian radial basis functions (RBF) $exp(-\gamma\|x_i - x_j\|^2)$ and sigmoid $tanh(\gamma x_i \cdot x_j + C)$. We have selected RBF kernels for our

[2] An odd is defined as the ratio of $P(Y=1|X=x)$ the probability of failing given the predictors X over $P(Y=0|X=x) = 1 - P(Y=1|X=x)$ the probability of not failing given X.

[3] Since a probability is obtained, it is still necessary to define a threshold to classify the outcome as 1 or 0. A method to identify the best threshold is given in Section 4.1.

[4] While the initial $y_i \in \{0,1\}$, the relabeled $y_i \in \{-1,+1\}$

analysis because they offer the best trade-off in terms of computing cost, stability and performance.

In our case, C-SVM with Gaussian kernels can be tuned by 2 hyperparameters:

1. C is a "cost" (i.e. regularization) parameter controlling over fitting: the larger the C, the higher the penalization of the error. It makes a compromise between the complexity of the model and the respect of the constraints in equation (3).

2. $\gamma = \frac{1}{2\sigma^2}$ is the scaling constant (aka kernel bandwidth parameter in non-parametric statistics) controlling the shape of the Gaussian kernel: the higher the γ, the smaller the standard deviation of the Gaussian.

3.2.3. Classification trees

A classification tree is another nonlinear non-parametric statistical technique consisting in a hierarchy of nodes obtained by recursive partitioning of the initial dataset. Each child node is characterized by a subset of its parent node[5] and is obtained by splitting the parent subset over a unique predictor, according to a threshold (continuous predictor) or partition over its levels (categorical predictor). Popularized by Breiman, Friedman, Stone and Olshen (1984), CART (Classification and Regression Tree) is the most popular implementation algorithm for classification trees and requires three elements:

1. a criterion to select the best dichotomic split at each node by minimizing a measure of error, typically the Gini index or a measure of information entropy
2. a stopping rule to decide whether a node is final - becoming a "leaf" of the tree – or whether the splitting process should continue
3. a decision rule to assign each leaf to a class (i.e. outcome) of the output Y.

The tree is progressively grown in a recursive fashion by dichotomic splits at each node until all leaves have been generated. Each leaf corresponds to one of the disjoint partitions of the initial dataset and is characterized by a simple model that differs from leaf to leaf. The full tree being often prone to overfitting, it is possible to "prune" it and obtain a smaller tree with less leaves but better performance. The mathematical formulation of a classification tree is relatively simple:

$$\hat{f}_\theta(X) = \sum_{m=1}^{J} \delta_m I\{X \in R_m\} \tag{4}$$

where J is the number of leaves of the tree, $I\{.\}$ the indicator function and δ_m the value of the class (i.e. 0 or 1) assigned to the m^{th} leaf corresponding to the subregion R_m of the 11-dimensional space of predictors. It should be noted that the variable names in Figure 2 and Figure 3 are ordered in a different manner than in Table 1.

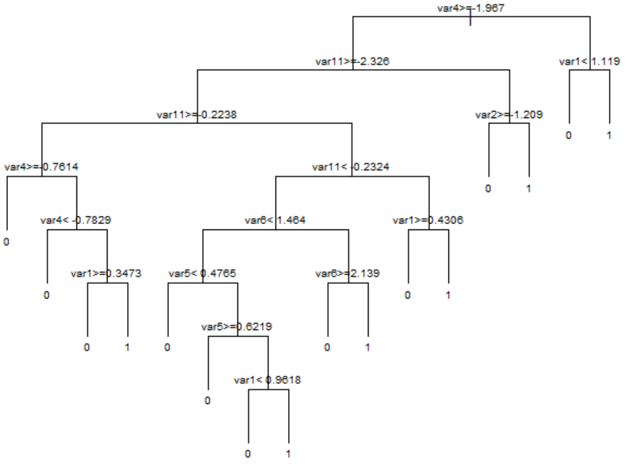

Figure 2 - Tree fitted on the case study (renamed variables)

Trees are particularly flexible since they accept indistinctly continuous, ordinal or binary predictors. They are also very easy to interpret thanks to the visualization of the tree structure (Figure 2). Last but not least, calculations on trees are particularly fast.

However, they are characterized by low bias and high variance: the addition of an outlier or a new observation in the dataset may dramatically modify the thresholds for the dichotomic split and lead to trees with very different classification results. A solution to this instability consists in "averaging" the predictions from a set of trees: this is the idea behind random forests and gradient boosted trees. As a consequence, although a single tree is a relatively weak binary classifier, it is actually a very important statistical technique as it constitutes the basis of more sophisticated models.

The choice of the splitting criterion being not a tuning parameter *per se* but rather a methodological choice, the performance of classification trees can be adjusted by 2 hyperparameters:

1. The number of leaves in the tree, which is related to the depth of the tree and the degree of overfitting.

2. The cost complexity parameter C_p that defines the minimum benefit to be obtained in terms of model fit before a split should be attempted. It is notably used to prune the fully-grown tree.

3.2.4. Random forests

Formalized by Breiman (2001), random forests is an ensemble model constructed by combining a large number of bootstrapped trees after random sampling with

[5] The first "root" node of the tree corresponds to the full initial dataset.

replacement amongst the N observations of the training dataset (X, Y) and also after random sampling amongst the p predictors X at each node. The class assignment is made by the majority vote on the class membership of the output Y (classification case). By averaging from a large number of uncorrelated, unbiased but high-variance single classification trees, the variance is reduced and the prediction accuracy is improved. The mathematical formulation of random forests is less simple because of the combination of single trees.

Random forests are not prone to overfitting (Hastie et al., 2009) and are also robust to outliers, noise, unbalanced datasets and missing data. Fast to compute, they provide estimates of correlations between predictors, the level of prediction accuracy and an assessment of the importance of each variable (Figure 3).

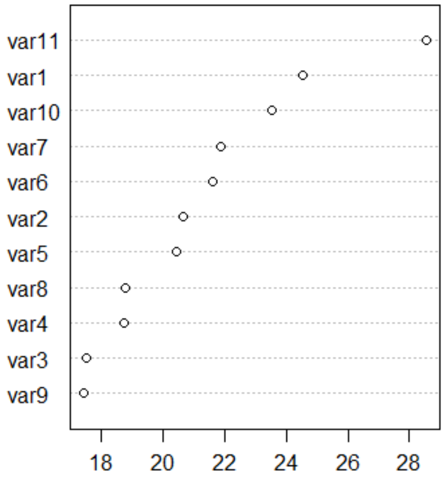

Figure 3 – Relative importance of the predictors

Random forests accept 3 main hyperparameters:

1. The number of single trees to be averaged into the random forest. The higher the number of trees, the higher the accuracy and the computing cost.
2. The number of predictors randomly sampled at each split.
3. The minimum size (i.e. number of elements) in the terminal nodes, or equivalently the maximum number of leaves in each of the individual trees.

3.2.5. Gradient boosted trees

Like Random forests, Gradient Boosted Tree (GBT) is an ensemble method based on combining a large number of single (classification) trees to form a stronger model. Contrary to random forests though, each individual tree in a GBT is weighted according to its prediction accuracy; a shrinkage parameter $0 < \nu < 1$ can also be defined to penalize the contribution of each tree when it is added to the GBT. Developed by Friedman (2001), boosted trees can be formulated mathematically as:

$$\hat{f}_\theta(X) = \sum_{m=1}^{M} w_m \sum_{j=1}^{Jm} \delta_{jm} I\{x \in R_{jm}\} \qquad (5)$$

where M is the number of trees while the weights w_m and the coefficients δ_{jm} are estimated by iterative procedures and are functions of the shrinkage coefficient ν.

Boosted trees can quantify the relative importance of the predictors as well as their nonlinear marginal influence (aka partial dependence) on the output Y. We showed in equation (2) that the coefficients β have a similar role in the logistic regression, although they were constrained to have a linear marginal influence.

Gradient Boosted Trees can be tuned by 3 hyperparameters, some of which are common to the hyperparameters of single trees:

1. The number M of individual trees to combine in the ensemble model, equal to the number of boosting iterations. The higher, the more accurate and the computing cost of the model. If M is too high though, over fitting might occur, contrary to random forests.
2. The size $J_{m\in\{1,\ldots,M\}}$ (i.e. the number of leaves) of each of the M constituent trees of the GBT.
3. The shrinkage parameter ν is penalizing each tree constituting a GBT. It is equivalent to the learning rate or the decay also encountered in neural network.

3.2.6. Neural networks

Neural networks are made of individual perceptrons whose output y_j can be written $y_j = f_j(\sum_i w_{j,i} x_{j,i})$ where $x_{j,i}$ are the inputs of the perceptron, $w_{j,i}$ its weights and f_j a so-called "activation function", typically the sigmoid $\sigma_s(x) = 1/(1 + e^{-sx})$. The perceptrons are organized in such a way that the output of a perceptron located upstream becomes the input of a perceptron downstream, forming *de facto* a net organized in three types of layers:

1. the input layer made of the p observed predictors,
2. the hidden layer(s) containing M non-observed perceptrons $Z_k = \sigma_s(w_{0,k} + \sum_{i=1}^{P} w_{i,k}\, x_i)$ computing the nonlinear features from linear combinations of the inputs
3. the output layer containing the probability of failure

In a neural network, the probability of each class of the binary output Y is therefore expressed as a complex nonlinear function of linear combination of the predictors:

$$P(Y = l \in \{0,1\}) = \frac{e^{\beta_{l,0}+\Sigma_{k=1}^{M}\beta_{l,k}\sigma_s(w_{0,k}+\Sigma_{i=1}^{P}w_{i,k}x_i)}}{\Sigma_{i=0}^{1}e^{\beta_{i,0}+\Sigma_{k=1}^{M}\beta_{i,k}\sigma_s(w_{0,k}+\Sigma_{i=1}^{P}w_{i,k}x_i)}} \quad (6)$$

Neural networks are very relevant to highly nonlinear problems and can produce very accurate results, despite being potentially subject to overfitting or non-convergence. However, they require a scaled dataset with no categorical predictors and a certain expertise in choosing the number of perceptrons and layers, the structure of the connections, the penalization (also called weight decay), amongst others. The two hyperparameters for neural networks are:

1. The decay ν is often compared to a learning rate, in the sense that it will penalize the estimation of the weights $w_{j,i}$ of the neural network
2. The maximum number of iterations before convergence. The higher this number, the higher the probability for the neural network to reach a stable and accurate solution

Each of the aforementioned binary classifiers exhibit advantages and drawbacks that have been extensively documented in Machine Learning literature (Huang et al., 2003). The next section presents a comparison of their merits through an application to our case study.

4. RESULTS

This section presents the results of models' performance, based on the methodology developed in Section 3. After a description of the criteria retained for comparing the models, we present an overall ranking of the binary classifiers, followed by a more quantitative assessment of model's performance based on two criteria: prediction accuracy and the c-statistic.

4.1. Criteria for comparing models

To account for the different characteristics of the aforementioned binary classifiers, we defined a set of comparison criteria:

1. The accuracy of the model is a quantitative criterion measured by metrics such as the percentage of outcomes correctly predicted or the area under the Receiver Operating Characteristic (ROC) curve, a typical tool in the field of machine learning applied to binary classification (Fawcett, 2006) (Figure 4).
2. The interpretability of the model is defined more subjectively as the difficulty to understand and use the results of the model for a subsequent engineering analysis.
3. The easiness to fit the model is a second qualitative criterion indicating the level of efforts and skills to actually train the model (selection of the predictor, tuning of the hyperparameters, etc.).
4. The cost is a quantitative assessment of the computing time needed to train the model. For a fair comparison, the measures are acquired on the same computer under similar conditions for all the models.

We decided to include two qualitative comparison criteria because the performance of a binary classifier can't be reduced to quantitative metrics such as accuracy or computing cost. The complexity in training, understanding and interpreting a model indeed represents a large hidden cost that might strongly limit the performance of the model and even prevent its use in some situations (low maintainability, poor formal training, lack of statistical skills of the users, etc.).

4.2. Overall comparison of binary classifiers

The 6 binary classifiers are compared according to the aforementioned criteria, each assesses on a qualitative scale in order to respect confidentiality agreement (Table 3). Each binary classifier presents advantages and drawbacks for each of the criteria.

Not surprisingly, ensemble models based on classification trees (random forest, gradient boosted trees) as well as other strongly nonlinear models (neural networks and SVMs to a lower extent) are much more accurate than the linear logistic regression or the unstable weak classifier (single tree).

Regarding interpretability, logistic regression provides the coefficients of the model, which allows estimating the marginal effect that each predictor has on the output. Single trees give an interesting visual view on the problem, provided the tree is not too deep (number of leaves smaller than 20). Random forests can rank the predictors according to their importance. The other classifiers are more difficult to interpret because 1) their mathematical formulation is not as easy and/or 2) they don't provide directly a measure of the influence of the predictors.

The easiest training and fit is obtained with robust models with few and conceptually simple hyperparameters such as decision tree, logistic regression or random forest and boosted trees to a lower extent. On the contrary, complex and unstable techniques such as SVM and neural networks require expertise to be properly trained and fitted.

Unsurprisingly, the more sophisticated and the higher the number of hyperparameters, the more computing resources are necessary to fit the model. There is almost a direct relationship between the easiness to train a model and its cost.

4.3. Focus on prediction accuracy of the classifiers

Even though qualitative criteria are important, the prediction accuracy is often attributed a greater importance when ranking models, as it might appear as the most objective criterion: the higher the prediction accuracy, the more likely the model will predict future outcomes with success. In the case of binary classifiers, the prediction accuracy is the number of outcomes correctly predicted (i.e. the sum of true positives and true negatives) over the total number of observations in the dataset.

Nonetheless, the prediction accuracy varies according to hyperparameters of the model or according to the cut-off threshold selected to separate positive ($Y = 1$) and negative ($Y = 0$) outcomes. This variation in accuracy is obtained by computing the prediction accuracy over a range of hyperparameters or cut-offs, whose results are visualized through the so-called ROC curve. The ROC curve expresses True Positive Rate (TPR aka sensitivity of the model) in the Y-axis as a function of the False Positive Rate (FPR equal to 1-specificity) in the X-axis (Figure 4).

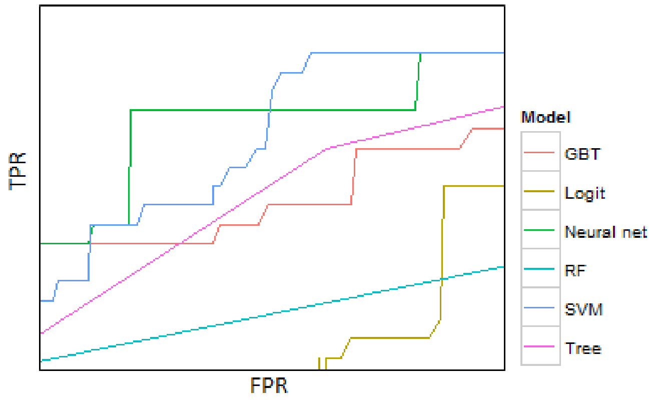

Figure 4. ROC curve comparing the classifiers' accuracy[6].

There are two efficient ways to evaluate the performance of a binary classifier from the ROC graph:

1. identify the point at the closest Euclidean distance from the top left corner of the ROC. This point corresponds to the highest TPR and the lowest FPR simultaneously, namely the highest prediction accuracy attainable by the model. This particular point has been retained as a common ground for comparing model accuracy, although Provost, Fawcett, and Kohavi (1998) debated over its robustness and relevance. To mitigate this effect, we generated Monte-Carlo simulations of 20 ROC by random sampling of training and test sets from the initial dataset, from which we extracted the worst, average and best prediction accuracy (Table 2). The averaged 20 simulations are presented in Figure 4 and allows a comparison between the 6 binary classifiers.
2. the Area Under the Receiver Operating Characteristic (ROC) curve (AUC), also called the c-statistic, corresponds to the "probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance" (Fawett, 2006). The c-statistic measures the quality of the classification of the binary classifier over the full range of a parameter or threshold. For this reason, it is often described as more robust at quantifying the average performance of a classifier than the mere prediction accuracy.

We compared the binary classifiers by computing their average accuracies and the c-stastistic in percent (first value in each cell), as well as the minimum and maximum values (values in brackets) from the 20 Monte Carlo simulations (Table 2). To respect industrial confidentiality however, we provide data relatively to the average fit of the logistic regression model (marked in bold), as it is the simplest of the aforementioned binary classifiers. Since neural networks are known to be very sensitive to non-scaled datasets, we give the accuracy and AUC for the non-scaled (first line in each cell) and the scaled versions of the dataset (second line in each cell).

Table 2. Accuracy and c-statistic of the binary classifiers relatively to logistic regression.

Classifier	Relative accuracy	Relative c-statistic
Logistic regression	**100%** [86-111] 101% [88-111]	**100%** [90-109] 100% [91-106]
SVM	106% [25-128] 105% [27-132]	115% [108-122] 115% [109-120]
Classification trees	114% [82-131] 112% [80-136]	108% [98-116] 107% [100-116]
Random Forests	134% [131-136] 134% [131-136]	111% [103-118] 111% [101-122]
Gradient Boosted Trees	121% [102-132] 120% [99-131]	115% [104-127] 114% [105-120]
Neural network	115% [61-132] 127% [119-132]	115% [77-126] 123% [115-129]

Several insights arise from Figure 4 and Table 2:

- The performance of the logistic regression is indeed the lowest on average, as measured by the prediction accuracy and the c-statistic. It is followed by classification trees, SVM, Gradient Boosted Trees, Neural Networks and random Forests, in that order.
- According to the partial ROC curve, some models are more accurate for some values of FPR and TPR. In absolute terms, neural networks are the most accurate for low to middle FPR while SVMs are more accurate for middle to high values of FPR.
- The variation in performance for different simulations of the same model is important, as measured by the range between the minimum and maximum values of

[6] The TPR and FPR normally vary from 0 to 1 in a ROC curve. Figure 4 is only an extract with unscaled axes from the full ROC curve as it serves only as an illustration of the principle of ROC curves.

the accuracy and the AUC. For instance, SVMs have a good average performance (accuracy=106%, AUC=115%) but a change in the dataset can lead to very poor (25%) or excellent (132%) prediction accuracies.

- The variation in model performance can be large for one criterion and not for the other. The lower the variation, the more robust the method and thus the higher degree of confidence one can have on the quality of the output of a given model. Again, SVMs offer a good illustration of this effect, as the accuracy has a high variance compared to the AUC
- Except for 8 out of 24 cases, scaling the dataset improves the prediction accuracy of the c-statistic. It is particularly significant for neural networks, whose lowest performance becomes one of the highest amongst all models.

5. DISCUSSION AND CONCLUSION

First of all, the results appear promising compared to the state-of-the-art, although confidentiality agreement impeded us to provide absolute performance of the binary classifiers.

The overall comparison of the binary classifiers shows that the models are complementary. As often in statistical modelling, there is no “one size fits all” but rather models whose dissimilar characteristics make them more suitable to different objectives or users. On the one hand, the logistic regression will be more adapted to an infrequent user with less statistical skills and interested in quickly obtaining an approximate estimate from a simple and robust model. On the other end, neural networks might be a better choice for a well-defined objective where high and robust prediction accuracy is required (e.g. the integration into an optimization system). Business objectives will decide on the trade-offs between the conflicting criteria in Table 3, knowing that accuracy is often the criterion against which the other criteria - interpretability, computing cost, easiness to fit- have to be traded with. Nonetheless, ensemble models based on trees – namely random forests and boosted trees – seem to offer a proper overall compromise: they are robust, easy to train and fit, not too costly for the performance increase they allow while still yielding deep insights if interpreted correctly.

A quantitative ranking of the models is somewhat arbitrary as the performance might not increase for the accuracy and the AUC simultaneously. Moreover, some models are more performing for some zones of the ROC curve, meaning that different binary classifiers should be chosen according to the target values of FPR and TPR. Thus, it might be worthwhile to create an ensemble “meta-model” based on a combination of the 6 models, eventually applied selectively to right portions in the dataset.

Variation in performance can be quite high and depends on two main factors, whose relative influence on the model robustness is challenging to assess:

1. The structure of the training and test sets randomly generated at each simulation. In such case, the absolute robustness of the models should be clearly questioned and the model should not be used, as it might not be possible to ensure the degree of accuracy of its predictions. SVM and to a lower extent single classification trees should thus be used carefully in our case study.
2. The internals of each method have some influence on the model performance: random generation of initial weights for neural networks, a local instead of a global minimum encountered by an optimization technique, etc. In such a case, the robustness of the model can be improved by tuning its hyperparameters. Nonetheless, this operation requires high statistical expertise and might not improve significantly the performance of a model

Scaling the initial dataset provides a better ground for comparing the models and almost always improves the model performance, should it be measured by the AUC or the prediction accuracy. This data transformation step is even necessary to ensure the relevance of neural networks, which finally ranks as the most performant in absolute terms. Thus, we recommend scaling the datasets whenever possible before fitting binary classifiers.

Next steps for future research can be formulated:

- The first step would consist in increasing the robustness of the performance assessment by generating more simulations (hundreds or even thousands) and taking quantiles or confidence intervals from the simulated ROC instead of the minimum and maximum values
- Improving the performance of each model might be a second step, done by better tuning of the hyperparameters and by adding more predictors, at the expense of a higher computing cost and probably for a marginal gain in performance.
- Compare the prediction accuracy of the statistical models with the manual engineering-based estimates done by seasoned maintenance engineers. This task would be time-consuming and uncertain, given the lack of structured data.

ACKNOWLEDGEMENT

The main author thanks the Portuguese Foundation for Science and Technology (Fundação para a Ciência e a Tecnologia - FCT) that financed this research under grant SFRH/BD/ 51121/2010. The authors also thank Rolls-Royce PLC for its support, for providing the case study and for ensuring that no confidential information had been introduced in this article. In particular, employees from Cost Engineering (Julie Cheung) and Service Engineering (Fay Bayley, Winfried Friedl) are acknowledged for their support and suggestions.

NOMENCLATURE

N	number of observations in the sample
p	number of predictors in the model
Y	$N \times 1$ output vector to be predicted, containing the probability or the occurrence of failure
X	$N \times (p+1)$ matrix of predictors (incl. intercept)
ε	$N \times 1$ vector of residuals of the model
β	$(p+1) \times 1$ vector of model's coefficients
f_θ	actual function explaining Y according to X
$\hat{f}_\theta$	estimate of the actual function

REFERENCES

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8

Caesarendra, W., Widodo, A., & Yang, B.-S. (2010) Application of relevance vector machine and logistic regression for machine degradation assessment. *Mechanical Systems and Signal Processing*, 24(4), pp. 1161-1171. doi:10.1016/j.ymssp.2009.10.011

Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 27.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, vol. 27 (8), pp. 861-874. doi:10.1016/j.patrec.2005.10.010

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5): pp. 1189–1232. doi:10.1214/aos/1013203451

Goebel, K., Saha, B., & Saxena, A. (2008). A Comparison of Three Data-Driven Techniques for Prognostics, *Proceedings of the 62nd Meeting of the Society For Machinery Failure Prevention Technology (MFPT)* (119-131), May 6-8, Virginia Beach, VA.

Grogger, J. & Carson, R. (1991). Models for truncated counts. *Journal of Applied Econometrics*, 6(3), pp. 225-238. doi: 10.2307/2096628

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer Verlag

Huang, J., Lu, J., & Ling, C.X. (2003). Comparing naive Bayes, decision trees, and svm with auc and accuracy. *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, November 19-22, Melbourne, FL.

Jardine, A. K. S., Lin, D. & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical System and Signal Processing*, vol. 20, pp. 1483-1510. doi: 10.1016/j.ymssp.2005.09.012

Jordan, W.E. (1972). Failure Modes, Effects, and Criticality Analyses. *Proceedings of the Annual Reliability and Maintainability Symposium* (30-37), January 25-27, San Francisco, CA.

Kalbfleisch, J. D., & Prentice, R. L. (2011). *The statistical analysis of failure time data*. Hoboken, NJ: John Wiley & Sons, Inc.

Kim, H-E., Tan, A C. C., Mathew, J., Kim, E Y. H., & Choi, B-K. (2008). Machine prognostics based on health state estimation using SVM . In Gao, J., Lee, J., Ma, L., & Mathew, J. (Eds.) *Proceedings Third World Congress on Engineering Asset Management and Intelligent Maintenance Systems Conference* (834-845). Beijing, China

Lawless, J. F. (2003). *Statistical Models and Methods for Lifetime Data (2nd ed.)*. Hoboken, NJ: John Wiley and Sons, Inc. ISBN 0471372153.

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1), pp. 1-14. doi: 10.1080/00401706.1992.10485228

Meeker, W.Q., & Escobar, L.A. (1998). *Statistical Methods for Reliability Data*. Hoboken, NJ: John Wiley and Sons, Inc. ISBN: 978-0-471-14328-4

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In Shavlik, J. (Eds.), *Proceedings of ICML-98* (445-453). San Francisco, CA: Morgan Kaufmann.

Rafiee, J., Arvani, F., Harifi, A. & Sadeghi, M.H. (2007). Intelligent condition monitoring of a gearbox using artificial neural network. *Mechanical Systems and Signal Processing*, 21(4), pp. 1746-1754. doi:10.1016/j.ymssp.2006.08.005

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation, *1st International Conference on Prognostics and Health Management (PHM08)* (1-9), October 6-9, Denver, CO.

Vapnik, V. (1996). *The Nature of Statistical Learning Theory*, New York: Springer

BIOGRAPHIES

Jean-Loup Loyer received a MSc degree in Aerospace engineering from the Institut Supérieur de l'Aéronautique et de l'Espace Toulouse and Imperial College London in 2007 complemented by a MSc degree in computational statistics and econometrics from the University of Toulouse in 2012. Between 2007 and 2010, he worked as an analyst and project leader in the French Prime Minister office. In 2010, he started a PhD in Statistics applied to Mechanical Engineering and Industrial Management at the Instituto

Superior Técnico Lisbon, Portugal, in partnership with MIT and Rolls-Royce plc. The main objective of his research is to develop innovative tools supporting decision making and improving financial evaluations over the entire lifecycle of industrial products. In particular, he has been analyzing large volumes of real industrial data and developed advanced statistical models to predict the manufacturing and maintenance costs of jet engines. He is a member of various societies in the domains of statistics (Société Française de Statistique, Royal Statistical Society) and aeronautics (3AF – Association Aéronautique et Astronautique de France, Royal Aeronautical Society).

Elsa Henriques holds a Master and PhD in Mechanical Engineering from Technical University of Lisbon. She is an Associate Professor at Instituto Superior Técnico in the Manufacturing Technologies and Industrial Management scientific area, being responsible for programmes in the “Engineering Design and Manufacturing” graduations and post-graduations. Over the last 25 years, she has participated and/or coordinated several national and European R&D projects in collaboration with the industry (from tooling to aeronautics and automotive industries), mainly related to the management of complex design processes and manufacturing systems. She has been involved in the supervision of several PhD students performing industrially oriented research in the Mechanical Engineering and Leaders for Technological Industries (MIT Portugal initiative) doctoral programmes, involving European companies. She has a large number of scientific and technical publications in scientific journals as well as national and international refereed conferences. Until 2010, she was a national delegate in the 6th and 7th Framework Programme of the European Union in the “'Nanosciences, nanotechnologies, materials and new production technologies'” thematic area.

Steve Wiseall is a Team Leader in the Design System Engineering - Cost Methods group, Rolls-Royce plc, one of four corporate methods groups aimed at developing generic capabilities for use across the business. He has 25 years experience in Rolls-Royce mainly centred around either doing or leading capability acquisition programmes and technology transfer. He has research interests in Optimisation, Robust Design, Virtual Manufacturing, Design For Manufacture, Manufacturing Process Capability and more recently Cost Modelling and its integration into the Design process.

APPENDIX

Table 3. Overall qualitative comparison of binary classifiers according to four criteria.

Classifier	Accuracy	Interpretability	Easiness to train and fit the model	Computing affordability
Logistic regression	*	***	**	***
SVM	**	*	*	*
Classification trees	*	**	***	***
Random Forests	***	**	**	**
Gradient Boosted Trees	***	*	**	*
Neural network	***	*	*	*

The binary classifiers are ranked qualitatively according to four criteria, amongst which accuracy that is related to results in Table 2. The more asterisks, the better the classifier on the criterion. The qualitative ranking can be interpreted for two criteria:

- For “Interpretability”: *** corresponds to a classifier directly returning regression coefficients, so that the results can be easily interpreted by non-specialists (i.e. coefficients of a multiple linear regression giving the marginal effect of the predictor). ** corresponds to either an easy visualization of the results (classification trees) or the classifier’s ability to return the relative importance of the variables (random forests). * is given to “black-box” models for which engineering insights are difficult (gradient boosted trees) if not impossible to obtain (SVM, neural networks) from the results.

- For “Easiness to train and fit”: *** corresponds to a classifier that can be used by anyone with normal engineering skills can manage with less than one days training. ** is awarded to classifiers requiring approximately 1-2 weeks of specialized training. * is given to models requiring professional expertise.