Using different machine learning approaches to evaluate performance on spare parts request for aircraft engines

Antonio Capodieci[1,*], *Antonio* Caricato[1], *Antonio Paolo* Carlucci[1], *Antonio* Ficarella[1], *Luca* Mainetti[1], and *Cristian* Vergallo[1]

[1] Department of Engineering for Innovation, University of Salento, Lecce, 73100, Italy

Abstract. The Aircraft uptime is getting increasingly important as the transport solutions become more complex and the transport industry seeks new ways of being competitive. To reach this objective, traditional Fleet Management systems are gradually extended with new features to improve reliability and then provide better maintenance planning. Main goal of this work is the development of iterative algorithms based on Artificial Intelligence to define the engine removal plan and its maintenance work, optimizing engine availability at the customer and maintenance costs, as well as obtaining a procurement plan of integrated parts with planning of interventions and implementation of a maintenance strategy. In order to reach this goal, Machine Learning has been applied on a workshop dataset with the aim to optimize warehouse spare parts number, costs and lead-time. This dataset consists of the repair history of a specific engine type, from several years and several fleets, and contains information like repair claim, engine working time, forensic evidences and general information about processed spare parts. Using these data as input, several Machine Learning models have been built in order to predict the repair state of each spare part for a better warehouse handling. A multi-label classification approach has been used in order to build and train, for each spare part, a Machine Learning model that predicts the part repair state as a multiclass classifier does. Mainly, each classifier is requested to predict the repair state (classified as "Efficient", "Repaired" or "Replaced") of the corresponding part, starting from two variables: the repairing claim and the engine working time. Then, global results have been evaluated using the Confusion Matrix, from which Accuracy, Precision, Recall and F1-Score metrics are retrieved, in order to analyse the cost of incorrect prediction. These metrics are calculated for each spare part related model on test sets and, then, a final single performance value is obtained by averaging results. In this way, three Machine Learning models (Naïve Bayes, Logistic Regression and Random Forest classifiers) are applied and results are compared. Naïve Bayes and Logistic Regression, that are fully probabilistic methods, have best global performances with an accuracy value of almost 80%, making the models being correct most of the times.

1 Introduction

The Avionic is a very interesting sector where Artificial Intelligence (AI), a powerful tool to help humans make difficult tasks, can be used. AI can be applied where a wide variety of

[*] Corresponding author: antonio.capodieci@unisalento.it

data is available, just like in Avionic where a great amount of data is well suited for maintenance applications.

In literature, a lot of studies on Fault Diagnosis and Prognosis (FDP) are carried out to improve maintenance on condition in aircraft world. In these studies, data coming from engines' installed sensors can be used to perform the Remaining Useful Life (RUL) prediction of parts (particularly aircraft engines), in order to schedule an efficient maintenance plan for a fleet of engines and to optimize warehouse handling. For instance, in [1] a methodology for RUL prediction, with a condition-based maintenance strategy that uses a Bayesian and a change-point detection model, is developed in order to optimize maintenance scheduling, resources and supply chain management. In [2] a Bayesian approach is taken to exploit fleet-wide data from multiple assets to perform probabilistic estimation of Remaining Useful Life for civil aircraft gas turbine engines. In [3] the fusion of two classifiers, Support Vector Machine (SVM) and Adaptive Neuro-Fuzzy Inference System (ANFIS), integrated into a common framework, is utilized to enhance the fault detection and diagnostic tasks.

All these studies include capabilities that allow for detection and isolation of early developing faults as well as prediction of fault propagation, and they develop into two main areas:

- Prognostic and Health Management (PHM): it includes real-time systems with a capability of detecting, isolating and predicting incipient failures;
- Condition Monitoring (CM): it is an active research area whose task is to monitor the health of a system in order to avoid catastrophic failures. It includes built-in test and external test equipment, that are generally used for the task of fault detection and isolation. [4]

There are strong benefit evidences (costs reduction and risks mitigation) of moving to a totally condition based maintenance strategy, based on engines' sensor data.

Another way in which AI can be used for maintenance purposes is using data coming from maintenance activities in order to optimize warehouse costs. However, studies performed just using avionic warehouse data are not frequent as those performed with data coming from the great amount of sensors installed into an aircraft engine. In literature, an approach using warehouse data has been adopted in the automotive domain, where a study conducted at Volvo [5] aims to explore for which spare part types at Volvo Trucks there might be a correlation between fault codes and spare part demand, to explore the impact of using condition monitoring data on forecast accuracy and to examine potential effects on the forecast process. Results show that, basing forecast on condition monitoring data instead of demand information from the closest downstream supply chain actor, reduce inefficiencies in the supply chain.

Similar economic benefit can be obtained in an avionic workshop, optimizing engines removal plan and its maintenance work, engine availability at the customer and maintenance costs, as well as obtaining a procurement plan of integrated parts with planning of interventions and implementation of a maintenance strategy.

The purpose of this study is to analyse and define a new methodology approach to the process of maintenance, based on Artificial Intelligence. The maintenance theme is chosen because, as also explained in [6], fleet maintenance is very costly and improvements in maintenance logistics, reductions in unscheduled events (and their consequences), and improvements in operational efficiency, can have an enormous impact on reducing costs. For this reason, this study aims of making full use of the information available in the different areas of the manufacturing process, gaining knowledge of maintenance activities related to specific issues, with the goal for the company to manage more efficiently its warehouse, organizing its stocks in the most profitable way.

The way to reach this goal is the development of a Machine Learning (ML) approach that allows a company to predict spare parts repair state. A dataset of fleet maintenance is used to develop an algorithm based on Artificial Intelligence for prognosis or warehouse dimensioning, implementing, among existing approaches (model-based, data-driven and knowledge-based [7]), a data-driven one. This dataset consists of maintenance data from workshop or engine operations (that is the repair history of a specific engine type, from several years and several fleets) and contains information like repair claim, engine working time, forensic evidences and general information about processed spare parts. These data are used for a multi-label classification approach, in order to build and train, for each spare part, a Machine Learning model that predicts the part repair state (classified as “Efficient”, “Repaired” or “Replaced”) as a multiclass classifier does. Then, global results have been evaluated using the Confusion Matrix, from which Accuracy, Precision, Recall and F1-Score metrics are retrieved, in order to analyse the model performance and the cost of incorrect prediction.

2 Dataset and analysis

An avionic company has provided a proprietary warehouse and mechanical workshop database. It consists of records of about 363 commissions (repair orders) related to a single airplane engine type and its modules, from 2013 to 2019. These records contain data such as: the reason why an engine/module is under repair, the working time, the maintenance level it has been subjected, forensic evidences related to the inspected parts and their supply costs and time.

A Machine Learning model that allows the spare parts repair state prediction can be represented as a function y=f(x) where, input features are chosen among those that can influence the repair operations and that are available before inspection. In this instance, the most sensible input features are the repairing claim (the reason why the engine/module is sent to the repair depot), as it is directly related to which kind of parts are inspected, and the working time, as it can be considered as an index of wear. The repairing outcome is the output to be predicted.

Before ML is applied, a dataset pre-processing phase is performed. The first step of this phase is the dataset deterministic analysis. In the catalogue, there are 1218 spare parts, which, after inspection, can result as Efficient (E), Repaired (R) or Replaced (S). Parts always associated with same result (this is because spare parts can be subjected to standard repair operations) are excluded. So the dataset has been analyzed in a deterministic way and it has been discovered that:

1. 0.25% (3) of 1218 spare parts are always repaired;
2. 4.11% (50) of 1218 spare parts are always replaced;
3. 22.41% (273) of 1218 spare parts are always efficient;
4. 73.23% (892) of 1218 spare parts have different repair associated states.

There is no need to predict the repair state for parts on which the same operation is always performed, as it is yet known, so the dataset is reduced to the 73.23% (892) of 1218 spare parts, which take different repairing class values.

As second step of dataset pre-processing, all involved features are turned into a categorical type:

1. The repair claim input feature values are: A, B, C, D and TBD (To Be Defined). Each value represents an engine problem that is not specified at this stage, so claims are

masked into 4 fake labels (A, B, C and D) and TBD label (used when the problem for which the engine goes under repair is not known). These values are considered like classes and a numerical value is assigned to each one.

Table 1. Repairing claims feature numerical transformation.

Categorical Class	**Numerical Class**
A	0
B	1
C	2
D	3
TBD	4

Notice that “TBD” is treated as a class, like the others.

2. The working time input feature is a continuous feature, as it represents the number of the hours an engine worked, so it is turned into categorical by creating range of values through percentiles. Firstly, 4 percentiles are found.

Table 2. Working time percentiles.

Percentile	**Value**
20th	1259.4 hours
40th	1963.4 hours
60th	2489.8 hours
80th	3092.4 hours

Table 3. Working time feature numerical transformation.

Percentile Range	**Value Range**	**Class**	**Class Description**
0th – 20th	0 - 1259.4 hours	0	Very Low Working Time
20th – 40th	1259.4 - 1963.4 hours	1	Low Working Time
40th – 60th	1963.4 - 2489.8 hours	2	Normal Working Time
60th – 80th	2489.8 - 3092.4 hours	3	High Working Time
80th - …	3092.4 - … hours	4	Very High Working Time

3. The repairing outcome output feature is categorical, as it can take 3 possible values: E (Efficient), R (Repaired), S (Replaced). Each spare part, after an inspection, is assigned with one (or more) of these classes. So a numerical value is assigned to each class.

Table 4. Repairing outcomes feature numerical transformation.

Categorical Class	**Numerical Class**
E (Efficient)	0
R (Repaired)	1
S (Replaced)	2

2.1 Proposed methodology

As the repair status of multiple spare parts has to be predicted, multiple labels have to be considered, so the problem to face is a multi-label classification problem.

As explained in [8], multi-label classification is a generalization of multiclass classification, which is the single-label problem of categorizing instances into precisely one of more than two classes. In the multi-label problem, there is no constraint on how many classes the instance can be assigned to. In this instance, it means that the output label is not the single one, but there can be more labels which to assign a different class to. For this reason, each label represents a different classification task, thus the approach is to create an ensemble, that is using multiple models (as many as spare parts) in order to predict the repair status of the respective part.

In this instance, the *i*th of n model can be considered as a function like the following:

$$\{X_1, X_2\} \rightarrow \{y_E, y_R, y_S\} \quad for\ i\ = 1, \ldots, n \qquad \text{i)}$$

Where:

- $X_1 := RepairClaim$;
- $X_2 := WorkingTime$;
- $y_E := Probability\ to\ find\ efficient\ the\ ith\ spare\ part$;
- $y_R := Probability\ to\ repair\ the\ ith\ spare\ part$;
- $y_S := Probability\ to\ replace\ the\ ith\ spare\ part$;
- $n := number\ of\ spare\ parts\ whose\ status\ must\ be\ predicted.$

The model related to the *i*th spare part is fed with two inputs (the engine/module repairing claim and working time) and outputs the probabilities to assign the repair status to each one of repair classes (Efficient, Repaired, Replaced).

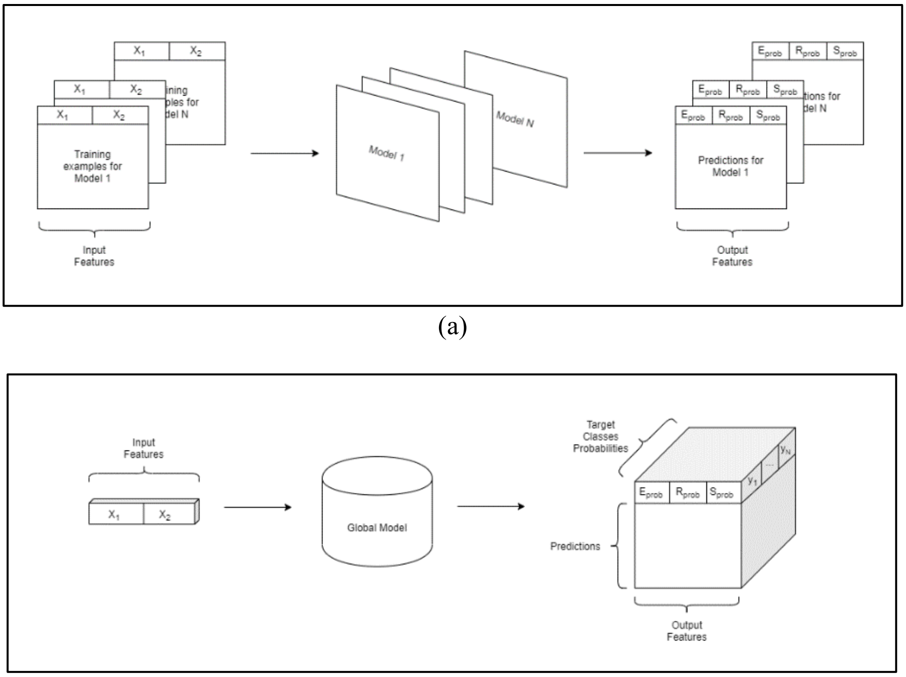
The training phase is performed using a different training set for each model, because the *i*th predictive model must be trained on those training examples that are related to commissions where the respective *i*th spare part is inspected. So the training for the *i*th model is performed considering the following set:

$$\bigcup_{j=1}^{c_i} \{X_1, X_2, y_i\}_j \qquad \text{ii)}$$

Where:

- $X_1 := the\ repair\ claim\ of\ commission\ j$;
- $X_2 := the\ working\ time\ of\ commission\ j$;
- $y_i := the\ repair\ outcome\ of\ ith\ spare\ part\ into\ commission\ j$;
- $c_i := the\ number\ of\ commissions\ with\ an\ inspection\ occurrence$ $for\ the\ ith\ spare\ part.$

Once the training examples related to the respective models are collected, the 30% of them is intended to be used as test set and so it is excluded from training. Then, the training phase is performed on each model independently on the remaining 70% of them (Fig. 1a).



(a)

(b)

Fig. 1. A model is built for each spare part. During the training phase, each model is trained on respective spare part training examples (a) and predictions are made for the related spare part. When all models are trained, they can be considered as a single system (b) making predictions for all spare part.

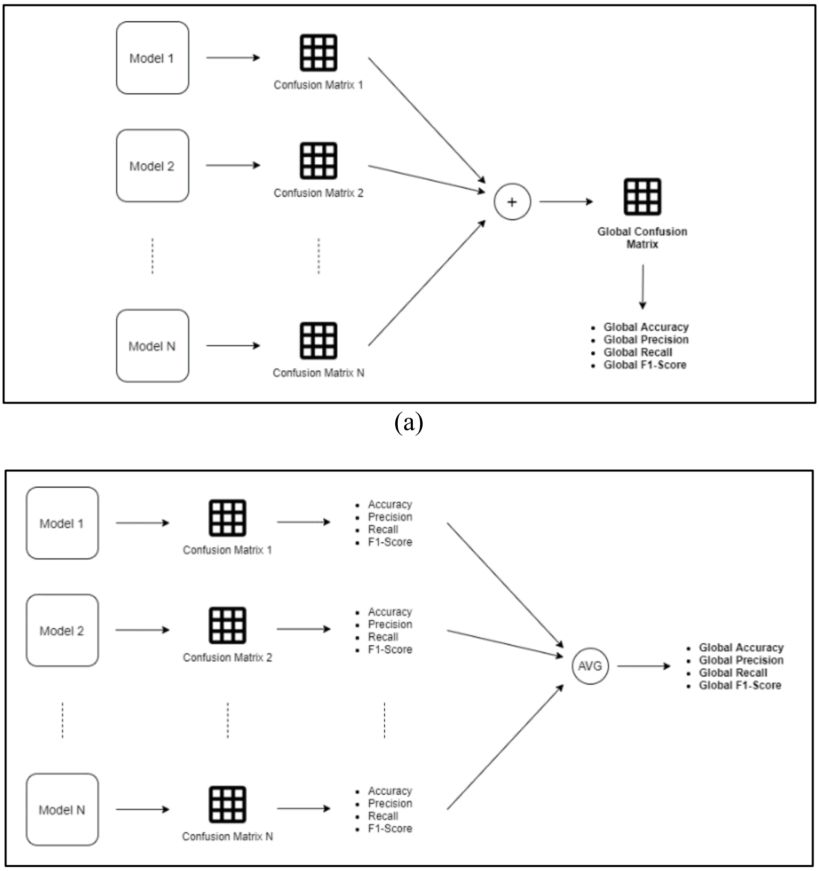
Once the training phase is performed, the result is a set of predictive models each one generating prediction for a specific spare part. This set of n models can be considered as a single global model (Fig. 1b).

The reason why the n models are considered like a single global one is that, firstly, starting from a specific commission (like a module commission that involves specific spare parts), once the input is fed, the output must be generated for each one of involved spare parts, so, even if it is the result of multiple models, the final output can be considered as generated from a single system. Secondly, in order to describe the models general performance, it would be preferable to have a single value related to a single system over multiple values related to multiple systems.

Evaluation metrics used to evaluate model performance are the Confusion Matrix, the Accuracy, the Precision, the Recall and the F1-Score metrics.

In order to get a single value for each performance metric, after the training phase, two evaluation approaches have been adopted:

1. Confusion Matrices are computed for each model and then they are summed up into a single matrix, respecting the classes cell order. On final Confusion Matrix, the remaining metrics are computed (Fig. 2a).
2. Confusion Matrices are computed for each model and respective metrics are computed, too. Then, final metrics values are obtained averaging each model-related metric (Fig. 2b).



(a)

(b)

Fig. 2. Scheme of the two evaluation methods adopted: (a) all Confusion Matrices computed for each model are summed up and then performance metrics are retrieved from it; (b) performance metrics are computed on each Confusion Matrix and then the single values are obtained averaging them.

Different classifiers are built through open-source tools provided by Scikit-learn Python library. Mainly, two classifier types are used: the first is purely probabilistic, like the Naïve Bayes (that is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions between the features) and the Logistic Regression (that uses a logistic function); the second is random-based, like the Random forest (that is an ensemble learning method that builds a multitude of decision trees at training time and outputting the class that is the mode of the classes).

3 Results and discussion

As previously said, for each *i*th spare part, a model is build: firstly, training examples related to the *i*th spare part are collected, then, 30% of them is used as test set and the remaining 70% are used into the training phase. So each spare part related model is trained on a different training set.

Results below are achieved on the test set, which is the set of unseen examples for models. This allows the model performance to be evaluated on data never seen by the model during the training phase. Once results for each *i*th model are obtained, they are merged together using the two evaluating approaches described before (Fig. 2).

The total number of samples which constitutes the test set is 8881. They are distributed among classes as plotted in the pie chart in Fig. 3. The number of actual cases in a class is 5790 for E-class, 851 for R-class and 2240 for S-class, thus a class imbalance can be noticed, as there is a prevalence of E-class (65%) on S-class (25%) and R-class (10%).

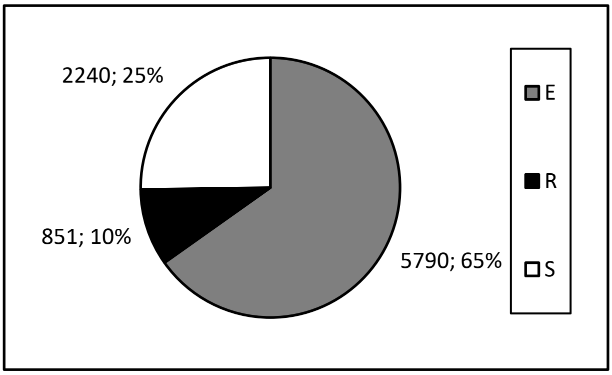
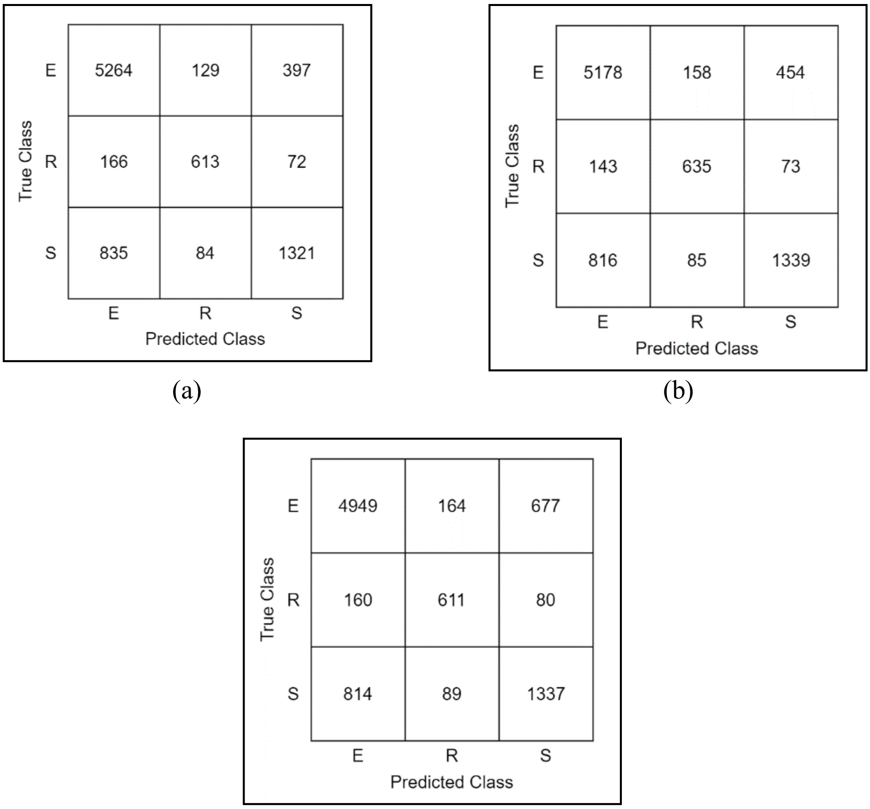


Fig. 3. Class distribution in the test set: E-class examples are 65% of total examples, R-class examples are 10% and S-class examples are 25%.

For evaluation method 1 (Fig. 2a), Confusion Matrices of each model related to the *i*th spare part are summed up, resulting into a single global Confusion Matrix for each classifier (Fig. 4), and once global Confusion Matrices are computed, other performance metrics (Accuracy, Precision, Recall and F1-Score) are computed, too.

Looking at Fig. 5, the number of cases classified as belonging to each class is extrapolated from Confusion Matrices reported in Fig. 4. In particular, for the E-class, there is an overestimation tendency as the number of actual cases is 5790 and all classifiers exceed this number (6265 for Naïve Bayes, 6137 for Logistic Regression, 5923 for Random Forest). Instead, for the R-class, which is not the prevailing class (in fact it is numerically the lowest), some classifiers overestimate its number of actual cases (851), like the Logistic Regression (878) and the Random Forest (864), while the Naïve Bayes underestimates it (826). Finally, for the S-class, there is a general tendency to underestimate its actual occurrences (2240), as the Naïve Bayes recognizes 1790 cases, the Logistic Regression 1866 and the Random Forest 2094.



(a)

True Class \ Predicted Class	E	R	S
E	5264	129	397
R	166	613	72
S	835	84	1321

(b)

True Class \ Predicted Class	E	R	S
E	5178	158	454
R	143	635	73
S	816	85	1339

(c)

True Class \ Predicted Class	E	R	S
E	4949	164	677
R	160	611	80
S	814	89	1337

Fig. 4. According with evaluation method 1, classifiers' Confusion Matrices are obtained summing up the Confusion Matrices related to each spare part model. This allows to consider a single matrix instead of a multitude for each classifier: Naïve Bayes (a), Logistic Regression (b) and Random Forest (c). Then these matrices are used to retrieve performance metrics.

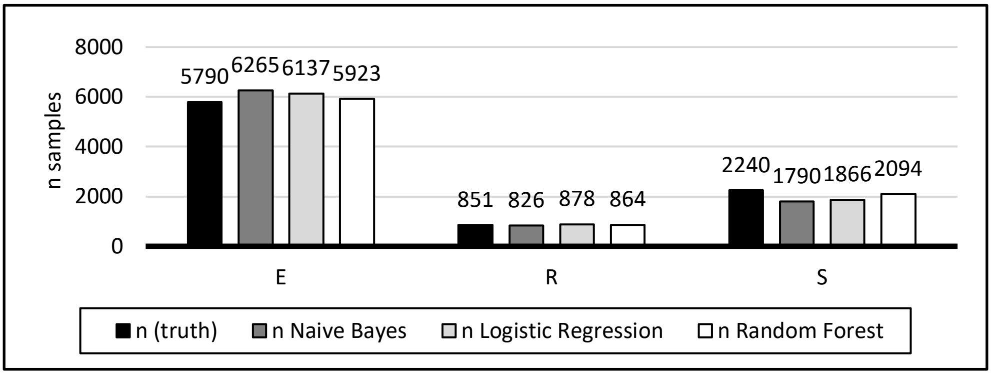
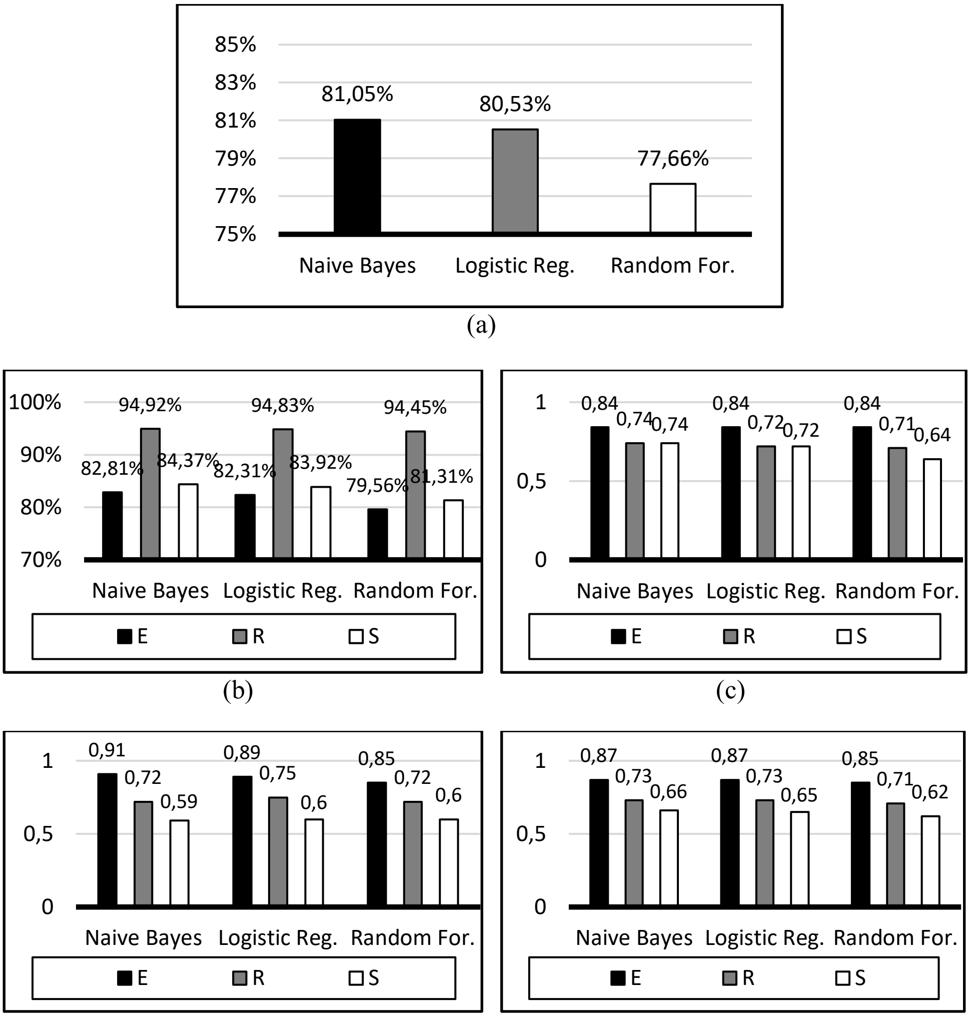


Fig. 5. Considering the global Confusion Matrices obtained through the evaluation method 1, for each classifier, the number of actual cases (truth) of each class is compared with the number of predicted cases (classified).

A deeper analysis is reported in Fig. 6, where, again, starting from Confusion Matrices reported in Fig. 4, classifiers' Overall Accuracy is represented in percentage form (Fig. 6a) and single-class performance metrics are compared among three classifiers (Fig. 6b-e).



(a)

(b)

(c)

(d)

(e)

Fig. 6. Once a single global Confusion Matrix is obtained with evaluation method 1, other performance metrics are computed. Firstly, Overall Accuracy (a) is computed as the percentage of correct predictions w.r.t. total samples. Then performances related to each class are computed: Single-Class Accuracy (b), Single-Class Precision (c), Single-Class Recall (d) and Single-Class F1-Score (e).

Classifiers' Overall Accuracy is computed as the percentage of correct predictions (the sum of diagonal in the Confusion Matrix) with respect to the number of total samples (the sum of each cell in the Confusion Matrix, that is 8881). In Fig. 6a, accuracies are compared and it shows that Naïve Bayes is the most accurate classifier, with 81.05% of Overall Accuracy, followed by the Logistic Regression (80.53%) and the Random Forest (77.66%).

Moreover, other performance metrics (Precision, Recall, F1-Score and Accuracy, too) in Fig. 6 are computed and compared with respect to each single class. This is done taking into consideration the multi-class problem as a binary classification problem where the reference class is analyzed considering when it occurs and when it does not occur (that is when one of other two classes occurs).

Considering the Single-Class Accuracy (Fig. 6b), the most accurate class to be predicted is the R-class with an accuracy of over 94% in all classifiers, this means that the R-class is detected most of the times. However, comparing all classifiers, the Naïve Bayes has best values for all classes with 82.81% of Accuracy for the E-class, 94.92% for the R-class and 84.37% for the S-class.

Moreover, considering the Single-Class Precision (Fig. 6c), the class with best values is the E-class, with values over 0.84. The term Precision quantifies the percentage of results which are relevant, as it is a value that indicates how many times the classifier is correct when it predicts a class. The classifier with higher Single-Class Precision results is again the Naïve Bayes with 0.84 for the E-class and 0.74 for R and S classes. Thus, when it identifies an E-class, it is correct 84% of the times while, when it identifies an R or S class, it is correct 74% of the times.

Single-Class Recall is another important metric as it refers to the percentage of total relevant results correctly classified by a classifier. The class with best values among all classifiers is the E-class, with values over 0.85, instead the class with worst values is the S-class, with values that never exceed 0.6. Here, the Naïve Bayes and the Logistic Regression can be considered both as the best classifiers. This is because for the E-class, the Naïve Bayes has a better value (91%) than the Logistic Regression (89%), but for the R and S classes, the Logistic Regression has higher values (75% and 60%) than the Naïve Bayes (72% and 59%).

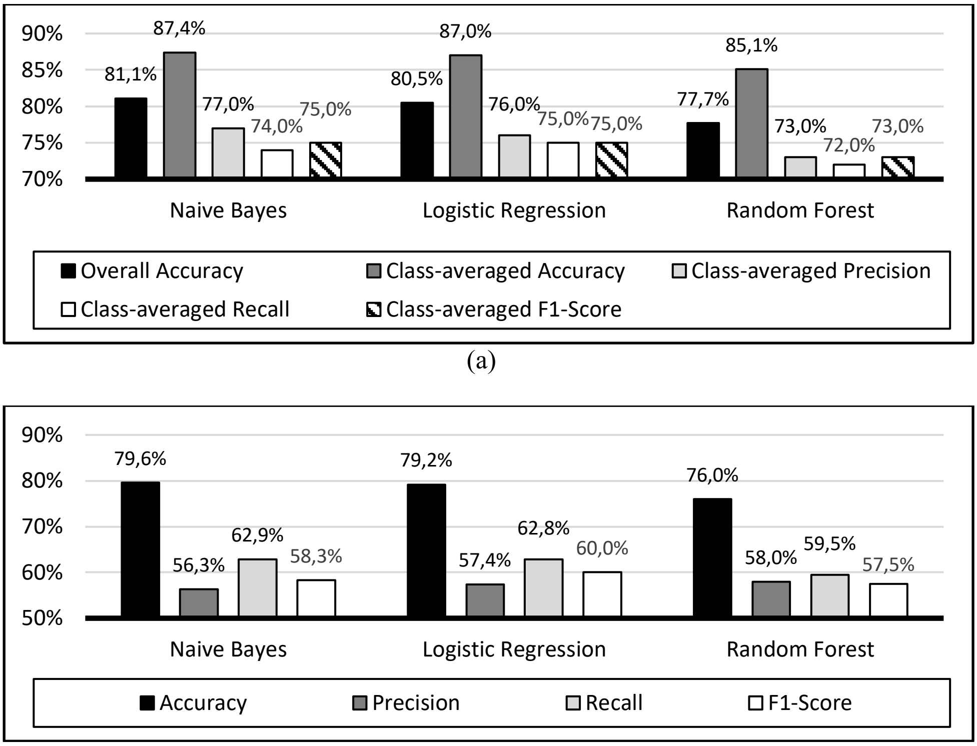
Finally, a combination of Single-Class Precision and Recall values is obtained considering Single-Class F1-Score metric. This metric confirms the goodness of Naïve Bayes with values of 87%, 73% and 66% for E, R and S classes.

Previously described values for all classifiers are reported in Fig. 6a, which shows that probabilistic classifiers have better performances.

For evaluation method 2 (Fig. 2b), Confusion Matrices of the model related to each *i*th spare part are used to compute performance metrics which are averaged resulting into a single set of values. Results are reported in Fig. 7b where probabilistic classifier results are again the best: the Naïve Bayes has the best Accuracy and Recall values (Accuracy is 79.6%, this means that almost 80% of predictions are correct, and Recall is 62.9%, it means that the model can identify almost 63% of correct classes); instead the Logistic Regression has the best Precision and F1-Score values (Precision is 57.4%, so when the model predict a class, it is correct 56.3% of the times, and F1-Score is 60%).

Results obtained with these two different evaluation approaches are different (Fig. 7). In particular, the first approach has higher values than the second approach. This is because, in the first approach, where a single global Confusion Matrix is obtained adding Confusion Matrices of each model, some important information, such those coming from class imbalanced datasets, can be lost. For instance, if a model has to predict E and R classes and it always outputs the E class, R class would have a recall of 0% but this information would be lost in a single global Confusion Matrix because on E/R cell other values are summed up, so it would be influenced by another model that has different recall for R class. In the second way, this information is not lost because each metric score affects the average (as it has a weight into the average), so metrics take into account class imbalance problems, too.

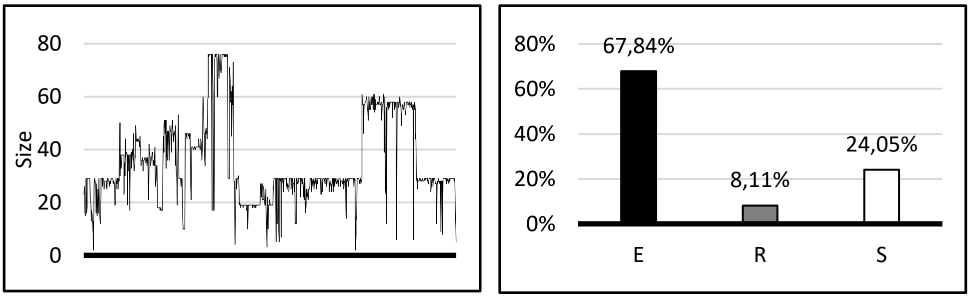
Even if in both evaluation approaches, probabilistic classifiers (the Naïve Bayes and the Logistic Regression) lead to better results with respect to the random-based classifier, results might have been even better if dataset had been larger and more varied. This is because the analyzed dataset has two main problems related to size and class imbalance, as said. As a different predictive model is built for each spare part and it is trained on its own training set, an analysis on dimensions and repairing classes of each training set is carried out (Fig. 9).



(a)

(b)

Fig. 7. Performances obtained with the first (a) and the second (b) evaluation method are compared among each classifier.



(a) (b)

Fig. 8. The dataset problem consists of a little number of training examples related to each spare part (a) and a class imbalance (b) in those training sets.

This led to discover that training set dimensions for the model related to each spare part can vary greatly, as each spare part has been worked a different number of times in the dataset. In particular, training sets have:

- Minimum dimension: 2;
- Average dimension: 34.97;
- Maximum dimension: 79.

Thus, on average, the training sets dimension is 35 and this is not a very large number for a training set. Moreover, there is a class imbalance problem because, analyzing the outcome repair state distribution within training sets, it is resulted that, on average, 67.84% of the times the training sets tend to have training examples for E-class, 8.11% of the times for R-class and 24.05% of the times for S-class. These two problems can be considered in model evaluation as responsible for not very high results and it is reasonable to believe that a larger dataset can contribute to higher performance.

In view of warehouse optimizing, the described ML approach can be very useful because it gives to an avionic company the predictive power of estimating which spare part needs to be replaced or repaired and so what it is necessary to be available in the warehouse. Indeed, in avionic sector, parts can be very expensive and their production can be time-consuming, so it is very important to organize the warehouse in a way that stock costs are reduced and, in the same time, availability of spare parts is high. This aspect is increased by downtime costs, that, are very high for aircrafts.

Moreover, the described approach can be used in a prognostic way: firstly making an estimate of expected number of commissions of the same type (e.g. with the same repair claim, working times, modules, etc.) in a given period and then using the ML predictive power to get the predictions on spare parts that are needed.

In this way, the company can see to it that its warehouse is well-stocked with necessary parts, shortening repairing time, providing a better service to customers and making them more satisfied.

4 Conclusions

In this work, an approach for avionic spare parts repair status prediction has been applied on an avionic company workshop dataset. On this dataset, it has been figured out that there is almost 30% of spare parts on which company does the same work for all commissions. This is because some parts do not undergo a physical degradation (thus, they result always as efficient) and others are subjected to degradation (thus, they are always repaired or replaced). Moreover, always replaced spare parts are discarded probably because this is a standard operation for those parts during maintenance process. Then, considering the remaining 70% of spare parts, on which different repair states are associated, three Machine Learning models have been applied, in order to predict their repair state starting from two information: the engine/module repair claim and the working time.

Results show that probabilistic classifiers, like the Naïve Bayes and the Logistic Regression, perform better than the random-based classifiers and that they can produce reliable results. Despite a dataset size problem, it can be concluded that developed ML models are useful tools to make spare parts repair status predictions as they can be used reliably in order to define the engine removal plan and its maintenance work, optimizing engine availability at the customer and maintenance costs.

5 References

1. M.A. Zaidan, R. Relan, A.R. Mills, R.F. Harrison, *Prognostics of Gas Turbine Engine: An Integrated Approach* (Exp. Syst. with Applic., Elsevier Ltd, 2015).
2. M.A. Zaidan, R.F. Harrison, A.R. Mills, P.J. Fleming, *Bayesian Hierarchical Models for Aerospace Gas Turbine Engine Prognostics* (Exp. Syst. with Applic., Elsevier Ltd, 2014).
3. K. Salahshoor, M. Kordestani, M.S. Khoshro, *Fault Detection And Diagnosis of an Industrial Steam Turbine Using Fusion of SVM (Support Vector Machine) and ANFIS (Adaptive Neuro-Fuzzy Inference System) Classifiers* (Energy, Elsevier Ltd, 2010).
4. H. Ren, X. Chen, Y. Chen, *Reliability Based Aircraft Maintenance Optimization and Applications* (Academic Press, 24th March 2017).
5. A. Ottosson, O. Wireklint, *Forecasting of Spare Parts Based on Vehicle Condition Monitoring Data - A Case Study at Volvo Group* (Department of Technology Management and Economics, Division of Supply and Operations Management, Chalmers University of Technology, Gothenburg, Sweden 2018).
6. A.J. Volponi (Pratt & Whitney, East Hartford, CT, USA), *Gas Turbine Engine Health Management Past, Present and Future Trends* (2005), Proceedings of ASME Turbo Expo 2013: Turbine Technical Conference and Exposition, June 3-7, 2013, San Antonio, Texas, USA.
7. M. Kordestani, M. Saif, M.E. Orchard, R. Razavi-Far, K. Khorasani, *Failure Prognosis and Applications - A Survey of Recent Literature* (IEEE Transactions on Reliability , IEEE, August 2019).
8. G. Tsoumakas, I. Katakis, *Multi-label classification: An overview* (Dept. of Informatics, Aristotle University of Thessaloniki, Greece, 2006).