Machine Predictive Maintenance Classification

Golemi Xhesika 302766, Pagano Alessandro 302283, Piovesan Alessandro 306097, Tiburzi Fabio 302307, Zappador Enrico 304669

***Abstract*—This study aims to accurately predict machinery breakdown by addressing the challenges presented by an imbalanced dataset. The main objective was to develop an accurate and reliable solution to identify failures in advance. To evaluate performances, we used common measures in the field of data engineering, such as f1 score, recall, and accuracy. The f1 score was chosen to provide a comprehensive evaluation of the model by combining the recall and precision metrics into a single meaningful value. In order to improve data quality, we applied several preprocessing techniques. To enhance data quality, various preprocessing techniques are applied, including handling inconsistent values, scaling and encoding features, and implementing class balancing methods. These operations were critical to mitigating the effects of data imbalance and enance model adaptability. Utilizing Python coding several models suitable for handling imbalanced datasets, including Support Vector Machines, Multilayer Neural Networks, Logistic Regression, and Random Forest, were tested iteratively with different data settings to identify the best performer. The experimental results showed that the proposed approach achieved promising performance in predicting machinery breakdown. The obtained macro average f1 score is 0.86 using Random Forest on the database modified with SMOTE. This results indicate the model's ability to properly balance recall and accuracy. This study may be of interest to industrial engineers and operators, providing tools to prevent failures and optimize plant maintenance, as the ability to identify machinery failures in advance is critical to improving the reliability and productivity of industrial plants.**

I. Introduction

The efficiency and dependability of industrial machines are greatly enhanced by predictive maintenance. Organizations may reduce expensive downtime, increase productivity, and optimize maintenance schedules by anticipating future faults and fixing them before they occur. However, given the proprietary nature of industrial operations, acquiring real-world predictive maintenance records is frequently difficult. In this study, we analyze a fictitious dataset that closely resembles actual industrial predictive maintenance scenarios. The following features are included in the dataset:

- UID: unique identifier .
- productID: consisting of a letter L(ow), M(edium), or H(igh) indicating quality and a number.
- Air temperature [K]: Generated using a random walk procedure, standardized to a standard deviation of 2 K about 300 K.
- Process temperature [K]: Generated using a random walk process, standardized to a standard deviation of 1 K. It is then added to the atmospheric temperature plus 10 K.
- Rotational speed [rpm]: Calculated using a power of 2860 W, overlaid with normally distributed noise.
- Torque [Nm]: Torque readings follow a normal distribution around 40 Nm with a standard deviation of 10 Nm. Negative values are not present.
- Tool wear [min]: Reflects the period of tool usage in minutes. The quality variants H/M/L contribute 5/3/2 minutes of tool wear, correspondingly.
- Failure label: indicates if the machine has been damaged.
- Failure Type: Indicates the type of cause of damage.

Using the available dataset, the main goal of this project is to create a binary classification model to forecast machine breakdowns. When a machine fails, it is indicated with the label "machine failure" for that particular data point. The categorization task entails establishing which of the following failure scenarios actually took place. To address this issue, our goal is to make use of the features already in place to train a predictive model that can precisely recognize the patterns and indicators of machine failure. Accurate failure forecasting enables proactive scheduling of maintenance tasks, reducing downtime and increasing operational effectiveness. In the following sections, we will examine different machine learning algorithms, perform feature engineering and selection, and assess the performance of each algorithm to choose the most efficient method for predicting machine failures.

II. Data analysis

In the early stages of analysis, we did a preliminary data validity study by implementing a filtering mechanism. Each instance in the dataset was analyzed to check for values outside the physical limitations outlined in the preceding section, like negative kelvin temperatures or tool wear minutes. This approach vouched for the correctness of the dataset, as all instances adhered to the defined limitations. We then took a look at the distribution of the 'Failure' column, which acts as the target variable for forecasting machine failure, in order to obtain insights into the dataset and aid in the interpretation of other factors. The 'Failure' column in the dataset showed a massive class imbalance, with only 3.19% of the rows resulting in failures. This is clearly a problem and needs further investigation, as it could lead to biased predictions, poor generalization, misleading evaluation metrics, and the loss of important information. The relationship between the "Failure" column and the "Failure Type" column was also something we decided to investigate, even though it was not the main subject of our inquiry. We found cases where the "Failure" column had a value of "0" but the matching "Failure Type" indicated a "Random Failure" and cases where the "Failure" column had a value of "1" but the "Failure Type" was marked as "No Failure." We decided to eliminate these cases since they caused a discrepancy between the two characteristics and threatened the dataset's integrity.
Subsequently, we did a complete study of the dataset by computing descriptive statistics to analyze the distribution

of features and identify any outliers. Upon reviewing the quartile values, we discovered a considerable variation between the third and fourth quartiles for these traits (Table I). To further analyze their distributions and locate outliers, we developed distribution charts and accompanying box plots for each variable. The box plots demonstrated the presence of outliers in both 'Rotational Speed' and 'Torque'(Fig 1, end of the section). This information is of great importance as it guides our upcoming steps in feature scaling, as we strive to find optimal algorithms that can properly manage such data outliers. These findings will contribute to the establishment of a robust and reliable model.

In order to examine the interrelationships between the characteristics and uncover probable patterns useful for feature selection and engineering processes, we ran a correlation study. This research involved generating a matrix that encompassed all conceivable combinations of the attributes and visually examining their associations. Additionally, the data points connected to different values in the 'Failure' column were color-coded to discover any observable trends in relation to failures.

The correlation chart demonstrated that 'Air Temperature [K]' and 'Process Temperature (K)' exhibited a substantial association (0.872474). Furthermore, an even stronger association was established between 'Torque [Nm]' and 'Rotational Speed [rpm]' (-0.878012) (Fig 2).

Regarding the relationship between failures and the characteristics, it was obvious from the chart that failures largely occurred at the extreme values of 'Torque [Nm]' and 'Rotational Speed [rpm]'. Specifically, failures were more prominent when one of these traits had a very low value and the other had a very high value. In contrast, no noticeable patterns or connections were detected among the other attributes, at least based on a visual inspection. These findings provide useful insights for additional feature selection and modeling considerations.

	AirT [K]	ProcT [K]	RPM	Torque	Wear [min]
count	4794	4794	4794	4794	4794
mean	299.98	309.98	1536.16	40.02	108.69
std	1.991	1.476	175.323	9.878	63.888
min	295.3	305.7	1168.0	9.3	0.0
25%	298.3	308.8	1423.0	33.3	53.0
50%	300.1	310.0	1504.0	40.1	109.0
75%	301.5	311.0	1607.75	46.7	164.0
max	304.4	313.7	2721.0	76.2	241.0

TABLE I
ATTRIBUTES DESCRIPTION

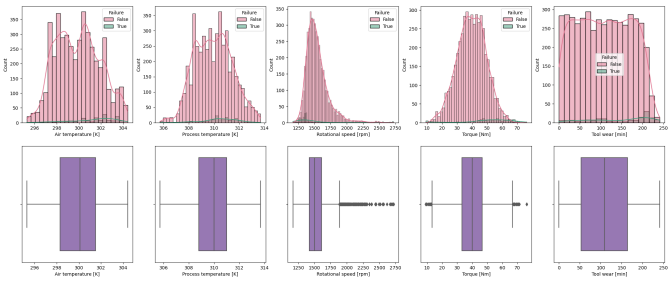

Fig. 1. Distributions and Boxplots of selected features.

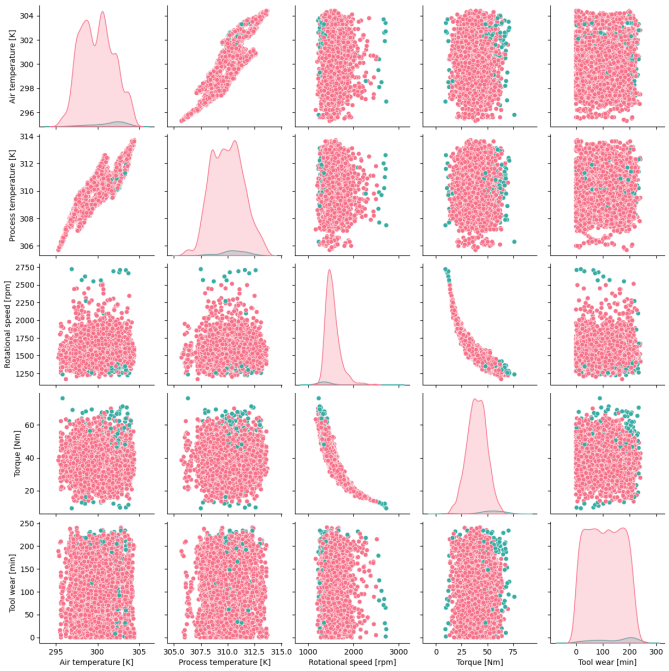

Fig. 2. Combinations of features to evaluate their correlation.

III. METHODOLOGY

We performed label encoding on the category variable associated with the tool quality before starting to develop any models since most methods demand numerical input data, which generally results in increased computing efficiency.

In order to prevent alghoritms to be biased toward features with higher values in magnitude we scaled some of the columns. In particularly for features containing outliers (such "torque [Nm]" and "rotational speed [rpm]"), the robust scaler is the preferred option since it is less influenced by outliers and scales the features using the median and interquartile range (IQR). We decide to apply the Min-Max normalization to other features.

Than we proceded testing the models using all the features at first and then gradually introducing feature selection to see if removing one or multiple features could be beneficial. To do this we tried to remove features that were highly correlated with each other, specifically one between 'Torque [Nm]' and 'Rotational Speed [rpm]' and one between 'Air Temperature [K]' and 'Process Temperature [K].' In this case, it is important to evaluate the trade-off between model efficiency and information loss related to their reduction. We find that, since the starting number of features is limited, removing one or more due to correlation is not beneficial for the model.

To lessen the imbalance of the data we experimented with using different oversampling approaches, in addition to selecting models that are more suited to work with unbalanced problems. We decide to employ SMOTE (Synthetic Minority Oversampling Technique) because it decreases the danger of overfitting by doing a synthetic replication of some data rather than randomly copying some instances like a Random Oversampler would do. Extreme caution was exercised in this situation since excessive oversampling can muddle some dataset trends and produce noise, resulting in the opposite of

what we are tying to achieve.

The first model we chose to test is Random Forest. During the training phase of the Random Forest, several decision trees are created using random samples extracted from the training dataset. This random extraction helps to maintain an appropriate balance between classes, allowing the model to learn from a wide variety of examples. An important advantage of Random Forest is the ability to assign different weights to classes when constructing decision trees. This means that it is possible to favor the recognition of the minority class by assigning it a higher weight.Due to the use of randomly drawn samples and the ability to assign different weights to classes, we expect the Random Forest to be effective in managing the balance between classes and aiding the recognition of minority classes during the training and prediction phase.

Secondly we opted towards. Support Vector Machines. SVM algorithm is a supervised learning method used for data classification. The main goal of SVM is to find the optimal separation hyperplane that maximizes the margin between training points belonging to two different classes. This is done by identifying a set of support vectors, which are the training points closest to the separation hyperplane. This algorithm is effective because, by finding an optimal margin, it can better generalize to new data not seen in training. In addition, SVM offers the possibility of assigning different weights to classes, again to better handle cases where classes are unbalanced.

Our third choice was Logistic regression. It uses a logistic function (or sigmoid) to estimate the probability that an instance belongs to a specific class trying to find the optimal coefficients that maximize the likelihood of the observed data. The output of the model is interpreted as a probability. In the context of highly unbalanced datasets, logistic regression can be useful because it allows handling of class skewness during training: the cost function penalizes errors on the minority class more. It also allows the decision threshold to be set manually to classify instances into the two classes, allowing the trade-off between sensitivity (recall) and specificity of the model to be managed.

Lastly we tried to implement a multilayer neural network utilizing Keras as interface for the Tensor-Flow library. Its structure is characterized by multiple layers that are organized in a linear sequence. Each layer, except the last, receives as input the output of the previous one and produces output that is passed as input to the next. Neural networks are able to learn complex representations of data and identify nonlinear patterns. This can be advantageous when it comes to classifying unbalanced classes, as they can capture nonlinear relationships between input features and the desired output. They can also use various regularization techniques to prevent overfitting, which is especially important in the case of unbalanced binary classification.

Having identified the models that we believe may be most efficient and chosen the changes we want to make to the data to improve performance, we iteratively apply the models to the various types of modified datasets and evaluate which matches are best. We mainly use precision, recall and f1 score for model evaluation, related to the minority (positive) class.

Recall is a measure that assesses the ability of a binary classification model to correctly identify positive examples. It is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN). Precision, on the other hand, assesses the proportion of positive examples predicted correctly relative to all examples predicted as positive. It is calculated as the ratio of true positives (TP) to the sum of true positives and false positives (FP). The f1 score, on the other hand, combines recall and precision into a single value, providing an overall assessment of model performance. It is calculated as the harmonic mean between recall (R) and precision (P). The f1 score is particularly suitable for balancing precision and recall in an unbalanced classification context. These measures provide a comprehensive assessment of model performance, taking into account both the accurate detection of positive cases and the correctness of positive predictions.

IV. Experimental results

The following results concern the models considered most suitable, tested on different datasets from the same database. Each modification was isolated to evaluate its effects. We present the results of the metrics related to the minority class, as they are of most interest. Our goal is to identify the best performing model and settings. Therefore, initially, we chose to test all models on all settings.

Initially, we evaluate the models on the default database, i.e., without modification except for the encoding of the categorical feature related to the type of machinery, in order to have benchmark for the models. To select the best parameters for each model we utilized Grid Search, a technique used in data analysis and machine learning to find the optimal set of hyperparameters for a given model, it involves defining a grid of possible hyperparameter values and exhaustively searching through all possible combinations to identify the best set.

	precision	recall	f1-score
Rand. F	1.00	0.42	0.59
SVM	0.62	0.42	0.50
LR	0.80	0.26	0.39
NN	0.83	0.32	0.47

TABLE II
TESTED MODELS ON DEFAULT DATABASE

Table II shows the scores for the selected metrics regarding the minority class achieved by the four algorithms in the benchmark scenario. It clearly shows that the random forest algorithm is the best in all aspects with these settings.

	precision	recall	f1-score
Rand. F	0.93	0.42	0.58
SVM	0.72	0.68	0.70
LR	0.67	0.19	0.30
NN	0.68	0.48	0.57

TABLE III
TESTED MODELS ON SCALED DATABASE

Table III shows the scores for the selected metrics regarding the minority class achieved by the four algorithms on scaled

database. SVM is the algorithm that improves the most as a result of database scaling.The normalisation of the data may have reduced the effect of features with different scales, allowing the SVM to learn the decision boundaries between classes more effectively. All metrics increase and we obtain a high recall score and f1-score (0.68 and 0.70). The neural network (NN) also follows the same improvement with lower performances. These modifications are not very effective for regression and random forest,this could be due to the fact that they are sensitive to the scale of the data.

	precision	recall	f1-score
Rand. F	0.83	0.65	0.73
SVM	0.51	0.58	0.55
LR	0.48	0.52	0.50
NN	0.61	0.45	0.52

TABLE IV
TESTED MODELS ON DATABASE OVERSAMPLED WITH SMOTE

Table IV shows the scores for the selected metrics regarding the minority class achieved by the four algorithms on database oversampled with SMOTE.The two methods that improve the most are Logistic regression and Random Forest. The latter in particular obtains a record f1-score of 0.73, while still maintaining high precision (0.83).The increase in data could have allowed Random Forest to better learn the patterns and characteristics of machinery failures, thus improving prediction performance. In contrast, SMOTE could alter the distribution of data, making it more difficult for the SVM to correctly learn the decision boundaries between classes.

	precision	recall	f1-score
Rand. F	0.79	0.61	0.69
SVM	0.45	0.55	0.49
LR	0.50	0.48	0.49
NN	0.61	0.55	0.58

TABLE V
TESTED MODELS ON DATABASE SCALED AND OVERSAMPLED WITH SMOTE

Table V shows the scores for the selected metrics regarding the minority class achieved by the four algorithms on database scaled and oversampled with SMOTE. Using both modifications at the same time has no advantage over using them separately. However, the results are better than in the benchmark case in which the database is not modified.

V. CONCLUSIONS

Maintenance planning and execution can be optimized with the developed model. The application can identify machinery at greatest risk of failure based on predictive models and suggest early preventive interventions to minimize negative impacts on production. Downtime would be reduced by taking prompt action before a complete failure occurs, reducing downtime and improving overall production efficiency. The consequence of these precautionary choices is to have an efficient allocation of resources such as personnel, equipment, and maintenance materials based on machinery failure forecasts, reducing costs and improving operational efficiency.

	Precision	Recall	F1	Support
0	0.98	0.99	0.99	712
1	0.83	0.65	0.73	31
Accuracy			0.98	743
Macro Avg	0.91	0.82	0.86	743
Weighted Avg	0.98	0.98	0.98	743

TABLE VI
COMPLETE RESULTS OF RANDOM FOREST WITHOUT SCALING, WITH SMOTE

	Precision	Recall	F1	Support
0	0.99	0.99	0.99	712
1	0.72	0.68	0.70	31
Accuracy			0.98	743
Macro Avg	0.86	0.83	0.84	743
Weighted Avg	0.98	0.98	0.98	743

TABLE VII
COMPLETE RESULTS OF SVM WITH SCALING, WITHOUT SMOTE

Finally, this information can support investment decisions and optimization of long-term maintenance strategies, enabling companies to allocate resources more efficiently and reduce overall costs.

Among the various models analyzed, we find that random forest trained on the database with SMOTE (where the minority class was raised to 12% of the total) is the best algorithm for the task at hand. In particular, to train this model, the number of estimators was set to 250, Gini Index was chosen as the criterion for splitting nodes, and the weight of classes was assigned to 1 for the main class and 1.7 for the minority class. This parameters were chosen after a series of iterative tests that led us to obtain the best results. In Table VI we report all the scores in detail.

Moreover, we consider highlighting the model obtained with SVM on the scaled database as it represents the maximum achieved recall (0.68) in terms of minority class. In this case we weight the minority class twice as much as the main one, the other parameters set are: 'gamma': 'auto', 'kernel': 'rbf'. We present these two models in order to give the domain expert the opportunity to evaluate which one is the most appropriate for the needs of the enterprise. This flexibility allows for a tailored approach to maximize the effectiveness of machinery failure prediction and preventive interventions, ultimately leading to improved operational efficiency and cost reduction.

VI. CONTRIBUTIONS

This was the first data analysis task for all members of the group, therefore we decided to work with an horizontal structure, trying to communicate as often, trading some time convenience with a diminished risk and lower pressure on individual choices. Individual preferences resulted in a different area of expertise for each member.

Golemi Xhesika 302766 : Models Research, Report Drafting; **Pagano Alessandro** 302283 : Exploratory data Analysis, Coding; **Piovesan Alessandro** 306097 : Coding, Neural Network Research; **Tiburzi Fabio** 302307 : Models Research, Coding; **Zappador Enrico** 304669 : Unbalanced Dataset Research, Report Drafting