

Machine Learning Models for Predictive Maintenance

Habiba HOUARI¹, Amina OUHOUD², Souaad TAHRAOUI³, Nada CHIKH⁴ and Bouchra ZAHF⁵

¹ Manufacturing Engineering Laboratory of Tlemcen (MELT)
Tlemcen, Algeria, Génie industriel Département National Polytechnic School of Oran, 31000 Oran Algeria
E-mail: habiba.houari@enp-oran.dz

² Manufacturing Engineering Laboratory of Tlemcen (MELT)
Tlemcen, Algeria, higher school of applied science Essa Tlemcen, Algeria
E-mail: ami_ouh@yahoo.fr

³ Electronic Département, Hassiba Benbouali University, Chlef Algeria, Signals ,Systems and Artificial Intelligence
Laboratory 2SAIL Chlef Algeria.
E-mail: s.tahraoui@univ-chlef.dz

^{4,5} National Polytechnic School of Oran 31000 Oran Algeria
E-mail: chikhnada13899@gmail.com

Abstract: To improve the management of the production line, we suggested two machine learning models to do the classification of failures in order to help the employees of the maintenance department in the management of the spare parts, through the results of measurements of performance evaluation of the suggested models, we were able to choose the best model that allows us to achieve our goal.

Keywords: Modelling, Simulation, decision making, predictive maintenance, machine learning.

1. INTRODUCTION

This next generation of information technologies has advanced over the few past years such as the Internet of Things (IoT), Big Data, Cloud computing, Artificial Intelligence (AI), etc., And their impact on the industrial sector is pushing the industry toward a smart industry or industry 4.0. the main focus of this essay is constructing a classification model, using Python for implementation, we suggest two classification models and then compare the results.

Predictive maintenance or maintenance 4.0 relies on using data that has been previously gathered and/or supplied by a computerized system (monitoring of infrastructures) to track the status of a machine or piece of equipment. it also enables periodic examination to identify the development of machine degradation and the need for technician involvement. in fact with the use of machine learning, and considering the information gathered, the use of the machine, and its history, predictive maintenance makes it easy to anticipate various kinds of malfunctions and issues

2. STATE OF THE ART ON ML FOR PREDICTIVE MAINTENANCE

The increase in data accessibility across practically all fields leads to requiring the use of algorithms for automated data analysis, which is pointed out by predictive maintenance which aspires to avoid system problems by continuously monitoring their state to schedule maintenance actions in advance.

The main obstacle that faces predictive maintenance is the analysis of the observation history in order to create predictive models. In this sense, machine learning has widespread cause it makes it possible to extract knowledge from a wide range of information sources with little to no human involvement.

For these reasons, numerous studies have been carried out (conducted) to utilize machine learning methods and strategies for industrial maintenance. for example (Yalcoue & Omar, 2021), They created predictive models in order to determine the life span of a motor, identify which motors are likely to fail in a given time frame and anticipate precisely when a motor will collapse

There are three categories of algorithms used to develop these models: regression, binary classification, and multiple classification. Using the proper performance evaluation indicators, they then confirmed the model's efficacy.

The author (Kreis, 2017) created a system for analyzing rotating machine vibrational signals in the time/frequency domain that was compared with a machine-learning system that identifies and categorizes flaws using intelligent algorithms. In (Arnaud, 2020), the author used methods such as k-means, support vector machine (SVM), and decision tree. They used a classification or confusion matrix and a "short score" to verify and compare the quality of the resulting models.

The author (Bouزيد, 2019) developed a system for analyzing the vibrational signals of a rotating machine in the time/frequency domain, which was compared with a machine learning system capable of detecting and classifying faults

using intelligent algorithms. The objective of the application was to offer a trustworthy fault detection system to minimize downtime and speed up the diagnosis of industrial system breakdowns. After running multiple tests the author was able to achieve a reliability of 99.3% and a classification rate of 100% for the various levels of failures considered.

In (Mifdal, 2019), suggested a brand-new prognostic approach based on the cloud computing model and the multitenancy principle to provide prognostics as a service. His method ensures better quality service while offering an efficient prognostic solution on demand. He used NASA aircraft engine data to construct and test three data-driven prognostic methods (artificial neural network, neuro-fuzzy system, and Bayesian network) in order to test the efficiency of his solution and compare the implemented methods.

In (Vijay, 2022), a state-of-the-art review of diagnostic and prognostic methods for failures based on data collected from equipment is presented, with the aim of avoiding the occurrence of a failure. According to the author, the main methods and tools using data are developed and concern their sources and processing, anomaly detection, principles of diagnosis and prognosis, residual life estimation techniques (DEFAD), and decision-making choices for maintenance strategies.

3. ML MODELS USED IN PREDICTIVE MAINTENANCE

Predictive maintenance is considered one of the very important elements of Industry 4.0. Hence, the demand to apply predictive analytics through ML models in order to predict failures and optimize asset management is rising in industries. Indeed, all these models operate on machine data in order to identify any trend or anomaly much before the failure happens. The choice of ML model may vary based on the nature of data, objectives, and specific industrial constraints.

3.1 Linear Regression

Linear regression is the most simple yet core model that is largely adopted to predict continuous values. It would, in simple terms, estimate the wear of equipment with regard to their historical operating data in predictive maintenance. Yet, this normally fails to capture the complex nonlinear relationship that characterizes industrial systems.

3.2 Logistic Regression

Logistic regression can be applied to a wide range of binary classification problems, such as predicting whether a given machine will fail in the near future. It is really applicable when the expected outcome of the variables is binary in nature, such as machine functioning or failure. Assigning a probability of failure to each machine, it helps in decision-making for maintenance scheduling.

3.3 Decision Trees and Random Forests

Decision trees are rule-based models wherein a target is predicted by iterating through a number of sequential questions regarding the features of the data. They are also

quite straightforward to understand and provide insight into the influencing factors in equipment failures.

It is extended by Random Forests: just forests of trees, really, improving robustness and accuracy in predictions. Applications of the model reach very wide arcs in predictive maintenance problems with big and complicated datasets and show impressive performance regarding accuracy and bias management.

3.4 Support Vector Machines

The support vector machines are classification algorithms that maximize the margin between different classes of data. For example, between the normal and faulty machine states, support vector machines are suited for complex data where separation of classes may not be easily feasible. Although data preprocessing is required, in general excellent performances can be achieved by SVMs, especially for small datasets.

3.5 Artificial Neural Networks (ANN)

Artificial neural networks model nonlinear relationships among variables by using inspirations from the human brain. ANNs process vast quantities of sensor data and can be used in Predictive Maintenance to detect early signs of failure. However, ANNs are not normally interpretable and often require enormous computational resources.

3.6 Learning

Deep learning-a more advanced form of neural networks-finds applications in general, increasingly in the field of predictive maintenance in such an environment when the amount of data received is huge and multidimensional. Deep architectures, like CNN and RNN, allow for sensor signals analysis and complex feature extraction for predicting failures. For example, CNNs can be applied to thermal images of machines, whereas RNNs are better suited for time series data, as often experienced in sensor data.

3.7 Ensemble Algorithms (Boosting, Bagging)

Ensemble methods combine a set of weak models to obtain a stronger overall model. The two most popular techniques are bagging and boosting: the bagging algorithm reduces overfitting by training several models on subsets of data, then combining predictions. Boosting algorithms successively improve the performance of an initial model by focusing on its errors. Those algorithms are suitable for complex and heterogeneous data, such as produced by several sensors in an industrial environment.

4. DATA COLLECTION AND PREPROCESSING METHODOLOGY

Quality and preparation of data most of the time define the accuracy of predictions from a machine learning model. Most commonly, data for predictive maintenance comes from sensors placed on machines, IoT systems, or even historical records.

4.1 Data Collection

Data for predictive maintenance typically comprises vibration, temperature, pressure, and other parameter measures that monitor the condition of the machines. The data can be in real-time or gathered periodically, as required. Strong systems of data collection are needed to be ensured that the ML models have information on reliable and representative conditions in reality.

4.2 Data Preprocessing

Data preprocessing is a must-have preparatory work before the introduction of machine learning models. It includes:

Data Cleaning: Handling missing values, clearing any duplicate entries, and removing inconsistent data. Data Normalization: The variable scaling of data to a similar magnitude scale to avoid dominance of one variable over others during training. Dimensionality Reduction: This refers to techniques like PCA that can help simplify a dataset while retaining important information.

4.3 Feature Engineering
 Feature engineering is a process of generating new relevant variables out of raw data in order to enhance the performance of ML models. Temporal derivatives are things like temperature changes or vibration that can be really useful and key for one to predict an imminent failure.

5. CONSTRUCTION OF THE CLACSSIFICATION MODEL

3.1 Presentation of tools Pythons

- Python: is a powerful and easy-to-learn open source programming language. Python is the programming language that has propelled itself to the forefront of infrastructure management, data analysis, and software development. Among its qualities, Python allows developers to focus on what they are doing rather than how they are doing it. It has freed developers from the constraints of form that occupied their time with older languages. Thus, developing code with Python is faster than with other languages.
- Google Colab or Colaboratory: is a cloud service offered by Google (for free), based on Jupyter Notebook and intended for training and research in machine learning. This platform allows models to be trained directly in the cloud, without the need to install anything on our computer except for a browser.
- Jupyter Notebook : an open source tool that allows users to write and share computer code for collaboration. Thanks to its many advantages, this tool has become an essential reference for Data Scientists.

3.2 Main modules and libraries used in our programs

- Pandas is a library that allows easy manipulation of data to be analyzed. It comes into play before the data set is prepared for training. Pandas facilitate

working with time series and structured multidimensional data for machine learning programmers.

- NumPy: this library allows for numerical calculations with Python. It introduces an easy-to-use way of handling arrays of numbers.
- Seaborn: an unparalleled visualization library, based on the foundations of Matplotlib. Storytelling and data visualization are important for machine learning projects, as they often require exploratory analysis of datasets to decide on the type of machine learning algorithm to apply. Seaborn provides a high-level dataset-based interface for creating amazing statistical graphics.

6. CHOICE AND DESCRIPTION OF SAMPLE DATA

The data samples used were extracted from the company's archives between January 13, 2019 and February 20, 2022. The data was confirmed by the maintenance department of the PVC tube unit, which allows our work to be based on reliable data. These data are grouped in an Excel file. (... lines) (See Appendix 2 Excel file).

The figure below represents an example of an Excel file that contains the description of breakdowns and the type of breakdowns (Mechanical, Electrical, Pneumatic, Hydraulic).

14	M	Scie fixation et serrage des supports tuyaux
15	M	Scie :Démontage , changement de courroie et remontage
16	E	Scie :Changement de support de charbon (LPV4)
17	M	Scie :Changement de disque à scie avec la confection d'une pièce
18	P	Scie :Démontage de silencieux , démontage de l'électrovanne , nettoyage et graissage et remontage
19	E	Scie : programme du compteur a été déréglé
20	P	Scie :Nettoyage de l'Electrovanne
21	M	Scie :Nettoyage de tambour
22	M	Scie :Ajout d'huile
23	M	Scie :Changement de disque à scie

Figure 1. Example of an Excel file.

6.1 Proposed Approach

In this section, we briefly discuss various algorithms proposed to model certain predictive maintenance objectives on the previously prepared dataset.

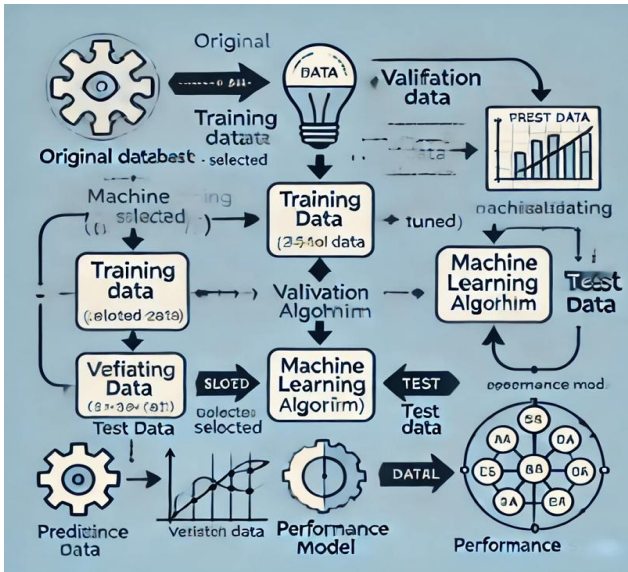


Figure 2. Steps to creating a predictive model.

6.2 Objective

The predictive maintenance objective dealt with in this project is: Multiclass classification: Predict the remaining lifetime of an engine before failure.

- *Multiclass classification.*

Multiclass classification applied to Predictive Maintenance can forecast two kinds of outcomes. It performs this by first allocating an asset to one of the several time periods in order to determine each asset's time interval of failure. Secondly, it shows the likelihood of future failure due to one of several root causes. This will enable maintenance staff to implement the problem in advance. Algorithms include logistic regression, decision trees, support vector machines, K-nearest neighbors, Gaussian naive Bayes, and random forest.

6.3 Model evaluation metrics.

Model performance evaluation is both highly critical yet a challenging task. Therefore, great caution should be taken to ensure the accuracy of the reported results.

It tells us how to evaluate the result of our model and which characteristics make one model better than another. For classification problems, several performance metrics have been proposed.

In a classification test, we classify items into two classes: positive and negative. It can be correct or incorrect. We draw this distinction by identifying four combinations of possible outcomes that may emerge for two classes: Positive and Negative. Possible Combinations of Outcomes(Yalcoue & all, 2021):

- True positive (TP): This is the total number of predictions in which the classifier actually correctly identifies the positive class as positive.

- True negative (TN): The total number of predictions wherein the classifier correctly identifies the negative class as negative.
- FP: Number of Predictions where the classifier has wrongly classified the negative class as positive.
- FN: The number of predictions where the classifier has incorrectly classified a positive class as negative.

a. *Accuracy*: it calculates the model's overall precision, or the fraction of the total number of samples correctly classified by the classifier.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

b. *Classification error rate*: it shows what percentage of predictions were incorrect. It is also referred to as classification error. It indicates how many of the predictions classified as positive were actually positive.

$$error\ rate = \frac{FP+FN}{TP+TN+FP+FN} \quad (2)$$

c. *Reminder*: It indicates the percentage of all positive samples correctly predicted as positive by the classifier. It is also called True Positive Rate (TPR) or Sensitivity.

$$Rappel = \frac{TP}{TP+FN} \quad (3)$$

d. *Specificity* indicates what percentage of all negative samples are correctly predicted as negative by the classifier. It's also known as the True Negative Rate (TNR).

$$Spécificité = \frac{TN}{TN+FP} \quad (4)$$

e. *The F1-score* combines precision and recall into a single measure. It is mathematically defined as the harmonic mean of precision and recall.

$$F1 - score = \frac{précision \times Rappel}{Précision + Rappel} = \frac{2TP}{2TP+FP+FN} \quad (5)$$

6.4 Confusion Matrix

A tabular method for presenting your prediction model's performance. Each cell of the confusion matrix gives the count of the predictions that the model made with class correct or incorrect.

a. *Confusion Matrix Binary Class*: In a binary classification problem, you have to classify just two classes-Positive and Negative.



Figure 3. Multi-class confusion matrix

b. *Confusion matrix multiclass*: Unlike binary classification, there is no positive or negative class in this. We have to find for each individual class TP, TN, FP, and FN.

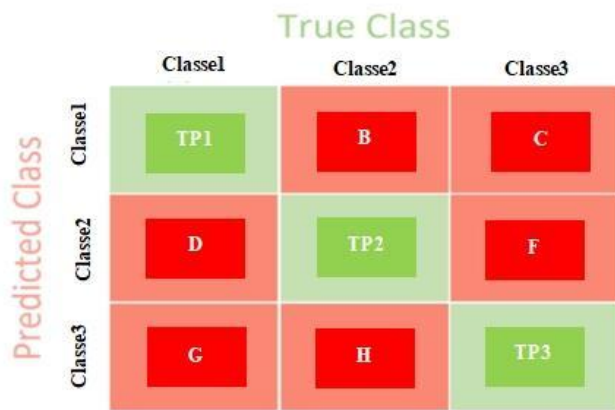


Figure 4. Multi-class confusion matrix

Classe1 :

- True Positive rate (TP) = TP1
- True Negative rate (TN) = TP2+F+H+TP3
- False Positive rate (FP) = B+C
- False Negative rate (FN) = D+G.

The dataset to be used for developing, testing, and evaluating our predictive maintenance approach is presented in this section. In the following section, we will work on model development using learning algorithms and performance evaluation using appropriate metrics.

Quantitative description of the data:

The summary state of the data is obtained by:

```
# Display dimensions of dataframe
print(data.shape)
print(data.info())

(187, 2)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187 entries, 0 to 186
Data columns (total 2 columns):
# Column          Non-Null Count  Dtype
---  -
0                 180 non-null    object
1  Description de la panne  180 non-null    object
dtypes: object(2)
memory usage: 3.0+ KB
```

Figure 5. Description of the sample breakdown data.

Below is a summary of missing data for each variable

```
data = data.dropna()
print("SUM : \n",data.isnull().sum())

SUM :
type                0
Description de la panne  0
dtype: int64
```

Figure 6. List of missing data from the breakdown sample.

Coding and counting of the factor variable (feature).

```
# Mapping of data feature values
data["type"] = data["type"].map({"E":1,"W":2,"P":3,"M":4})

# value_counts() function : Return a Series containing counts of unique values.
print("type : \n",data["type"].value_counts())

type :
4    88
1    59
2    28
3     5
Name: type, dtype: int64
```

Figure 7. Encoding and counting of factors.

- Test and Training Data: We choose to use 20% of the data for testing and 80% for training.

```
[ ] # split the dataset into training data and test data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(texts, y,
                                                test_size= 0.20, random_state=42, stratify=y)

print("X_train shape : ",X_train.shape, " y_train shape : ",y_train.shape)
print("X_test shape : ",X_test.shape, " y_test shape : ",y_test.shape)

X_train shape : (137, 257) y_train shape : (137,)
X_test shape : (35, 257) y_test shape : (35,)
```

Figure 8. Training and testing of data.

- Proposal of classification models and evaluation Accuracy: accuracy gives the overall precision of the model, i.e., the fraction of the total number of samples that were correctly classified by the classifier. Below, we propose two models for classification.
- The first proposed model for classification:

```
[ ] # training the model on training set
from sklearn import metrics
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
#from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression

# classifier = MultinomialNB()
classifier = LogisticRegression(random_state=42)

classifier.fit(X_train, y_train)

# making predictions on the testing set
predicted = classifier.predict(X_test)
```

Figure 9. Logistic regression classification.

Classification report :				
	precision	recall	f1-score	support
1	0.80	0.67	0.73	12
2	0.50	0.25	0.33	4
3	0.00	0.00	0.00	1
4	0.65	0.83	0.73	18
accuracy			0.69	35
macro avg	0.49	0.44	0.45	35
weighted avg	0.67	0.69	0.66	35

Figure 10. Classification report for the logistic regression model.

We can observe that the accuracy of the logistic regression model is 0.69.

- The second proposed model for classification: KNN (k-nearest neighbors).

```
[ ] from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
import seaborn as sns
# Create the classifier

knn_cls = KNeighborsClassifier(n_neighbors =6)

# Train the classifier
knn_cls.fit(X_train,y_train)

# Making predictions on the testing set
predicted = knn_cls.predict(X_test)

# Classification report
print("Classification report : ")
print(metrics.classification_report(y_test, predicted))
```

Figure 11. Classification by the KNN model.

Classification report :				
	precision	recall	f1-score	support
1	0.90	0.75	0.82	12
2	0.67	1.00	0.80	4
3	0.00	0.00	0.00	1
4	0.84	0.89	0.86	18
accuracy			0.83	35
macro avg	0.60	0.66	0.62	35
weighted avg	0.82	0.83	0.82	35

Figure 12. Classification by the KNN model.

We observe that the accuracy of the KNN model is equal to 0.83.

- Comparing the results of the models: we conclude that the KNN learning model presents better prediction results compared to the logistic regression model.

7. CONCLUSIONS

We have presented the various stages of design and implementation of our classification models.

To improve production line management, we have suggested two machine learning models for fault classification to assist maintenance service employees in managing spare parts.

To evaluate the machine learning performance of the proposed models, a comparative study was conducted between the logistic regression learning model and the KNN learning model by calculating evaluation metrics for both models. Based on the results, we have finally chosen the best model (the KNN model) that allows us to achieve better prediction results in our fault classification case.

This machine learning model is a decision support tool for predictive maintenance where predictive maintenance is a winning strategy for controlling spare parts inventory.

REFERENCES

Arnaud, W.(2020). Application des algorithmes d'apprentissage automatique pour la détection de

défauts de roulements sur les machines tournantes dans le cadre de l'industrie 4.0 . Thesis for the degree of master of applied sciences in engineering, university of Quebec at Chicoutimi, 69p.

Bouزيد, Z.(2019). Pronostic des systèmes industriels basé sur l'intelligence artificielle Maintenance prédictive . LMD doctoral thesis in computer science, Mohamed Khider University – Biskra, 153p.

Kreis, P.(2017). Apprentissage automatique pour la maintenance prédictive dans le domaine de l'Aviation". Doctoral thesis, University of Paris ,P.115.

Mifdal, R.(2019). Application des techniques d'apprentissage automatique pour la prédiction de la tendance des titres financier, L'obtention De La Maitrise. School of Higher Technology University of Quebec. p 176.

Vijay, K.(2022). What Is Machine Learning? Definition, Types, Applications, and Trends,<https://www.spaceworks.com/tech/artificial-intelligence/articals/what-is-ml/>

Yalcoue, F. & .Omar, A(2021). Machine learning pour la maintenance prédictive. end-of-study dissertation with a view to obtaining the Master's degree in industrial maintenance, P.71,.