Preventive maintenance of Propeller Using Machine Learning

A Study Applied to Mechanics and Numerical Modeling (MMN)

Ludovic Bocquillon

December 9, 2025

Abstract

This project focuses on the multiclass classification of rotor imbalance detected by vibration sensors using machine learning techniques. Its main goal is to build a pipeline capable of predicting the level of imbalance (from 0=no imbalance to 4=high imbalance) from raw features. During this project, we used and compared several classical and ensemble models, including Logistic Regression, Random Forest and XGBoost. We explore data imbalance, propose resampling solutions, tune model parameters, and evaluate our models using mainly accuracy and F1-score. The field of application of such a model is in clear relation with the MMN specialization. This project was chosen in the continuity of the SpectraV2 Project -development of test bench for drone propeller- of the Aerospatiale association “LeoFly”. In a vastly different context, with a different dataset, it could also be refined to be used on any test propeller test bench using other features.

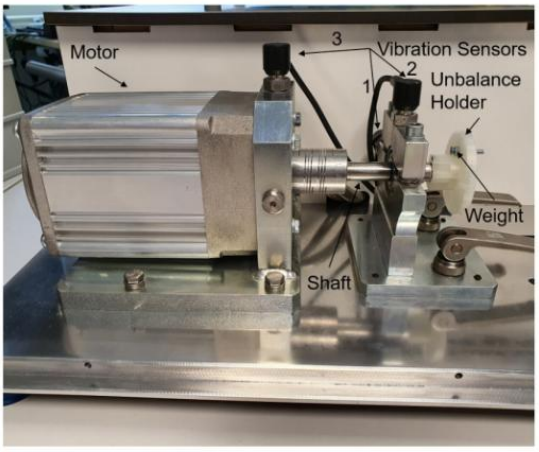
Table des matières

1 Business Case 3
2 Dataset Description 4
3 Data Exploration and Data Refinery 5
3.1 Data Visualization 6
4 Models Description & Limitation 10
4.1 Logistic Regression (raw data) 10
4.2 Random Forest (raw data) 10
4.3 XGBoost (with only temporal features including Skewness & Kurtosis) 10
4.4 XGBoost (with temporal and FFT features) and GridSearch for hyperparameter optimization 10
5 Obstacles and Solutions 12
5.1 Memory Limitations with Full Dataset 12
5.2 Frequency Domain 12
5.3 Time calculations 12
5.4 Overfitting and optimization 12
6 Results and Comparison 13
7 Conclusion 14
8 References 15

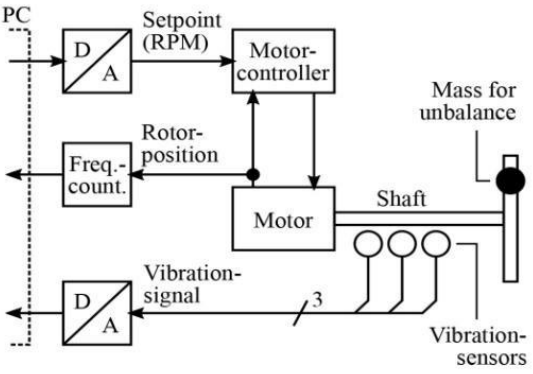
1 Business Case

Vibration is defined as a cyclic or oscillatory motion. Vibration analysis allows for the detection of defects in rotating machinery before they cause critical failures. The use of machinery vibration and the technological advances that have been developed over the years now make it possible to not only detect when a machine is developing a problem, but also to identify the specific nature of the problem for scheduled correction

The setup for the simulation of defined unbalances and the measurement of the resulting **vibrations** is powered by an electronically commutated **DC motor** (WEG GmbH, type UE 511 T), which is controlled by a motor controller (WEG GmbH, type W2300) and is fixed to the aluminum base plate by means of a galvanized steel bracket.



Measurement setup



Block diagram of the measurement setup

Vibration sensors (PCB Synotech GmbH, type PCB-M607A11 / M001AC) are attached to both the bearing block and the motor mounting and are read out using a 4-channel data acquisition system (PCB Synotech GmbH, type FRE-DT9837).

The objective of the project is to build a Machine Learning model capable of predicting the imbalance level (0-4) based on raw or transformed sensor data. Note that imbalance can represent a motor-blade wear out or even worse a broken motor blade. The ultimate goal of the project is to use this model on the data collected by LeoFly's engine test bench.

2 Dataset Description

Using the setup described in above section, vibration data for imbalances of different sizes was recorded. The vibration data was recorded at a sampling rate of 4096 values per second.

In total, datasets for **4 different unbalance strengths** were recorded as well as one dataset with the **unbalance holder without additional weight** (i.e. without unbalance). The **rotation speed** varied between approx. **630 and 2330 RPM**.

- 0 → rotor balanced (no added weight)
- 1–4 → increasing levels of unbalance, from small to severe

By varying the level of unbalance, different levels of difficulty can be achieved, since smaller unbalances obviously influence the signals at the vibration sensors to a lesser extent. Smaller imbalances are harder to detect, as they cause subtler changes in the vibration signals.

Each CSV file contains five features

1. **V_in**: Input voltage to the motor controller (V)
2. **Measured_RPM**: Rotor speed (RPM, computed from sensor measurements)
3. **Vibration_1**: Signal from the first vibration sensor
4. **Vibration_2**: Signal from the second vibration sensor
5. **Vibration_3**: Signal from the third vibration sensor

3 Data Exploration and Data Refinery

We analyzed:

- the number of rows and columns,
- presence of missing values,
- Outlying values.

```
dataset : 0D

Shape: (26423295, 5)

 Column info :
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26423295 entries, 0 to 26423294
Data columns (total 5 columns):
 #   Column        Dtype
---  ------        -----
 0   V_in          float64
 1   Measured_RPM  float64
 2   Vibration_1   float64
 3   Vibration_2   float64
 4   Vibration_3   float64
dtypes: float64(5)
memory usage: 1008.0 MB
/n
```

	V_in	Measured_RPM	Vibration_1	Vibration_2	Vibration_3
count	2.642330e+07	2.642330e+07	2.642330e+07	2.642330e+07	2.642330e+07
mean	5.992249e+00	-3.572305e+04	1.981370e-03	2.713846e-03	4.202591e-03
std	2.329660e+00	2.987905e+06	6.284235e-02	8.682009e-02	6.599321e-02
min	0.000000e+00	-2.400000e+08	-1.067495e-01	-2.067244e-01	-3.653050e-02
25%	4.000000e+00	1.052549e+03	-1.059771e-03	-1.188517e-03	1.739263e-03
50%	6.000000e+00	1.483102e+03	6.246567e-04	6.520748e-04	2.712011e-03
75%	8.000000e+00	1.906805e+03	2.528429e-03	2.536774e-03	3.825426e-03
max	1.000000e+01	2.376685e+03	7.805610e+00	8.780816e+00	7.790682e+00

Each CSV (0-4) has approximately the same number of rows (roughly 26 423 295).
There is no missing value, and all the values seem to be correct except for one feature, Measured_RPM, where the max and min values should be approximately between 600 and 2400 rpm.
We see that there are some absurd values such as -240 000 RPM and 3800 RPM
We will get rid of these outliers when the 5 datasets will be merged.

V_in	
0.00	12288
2.00	163840
2.05	163840
9.90	163840
9.95	163840
10.00	163839

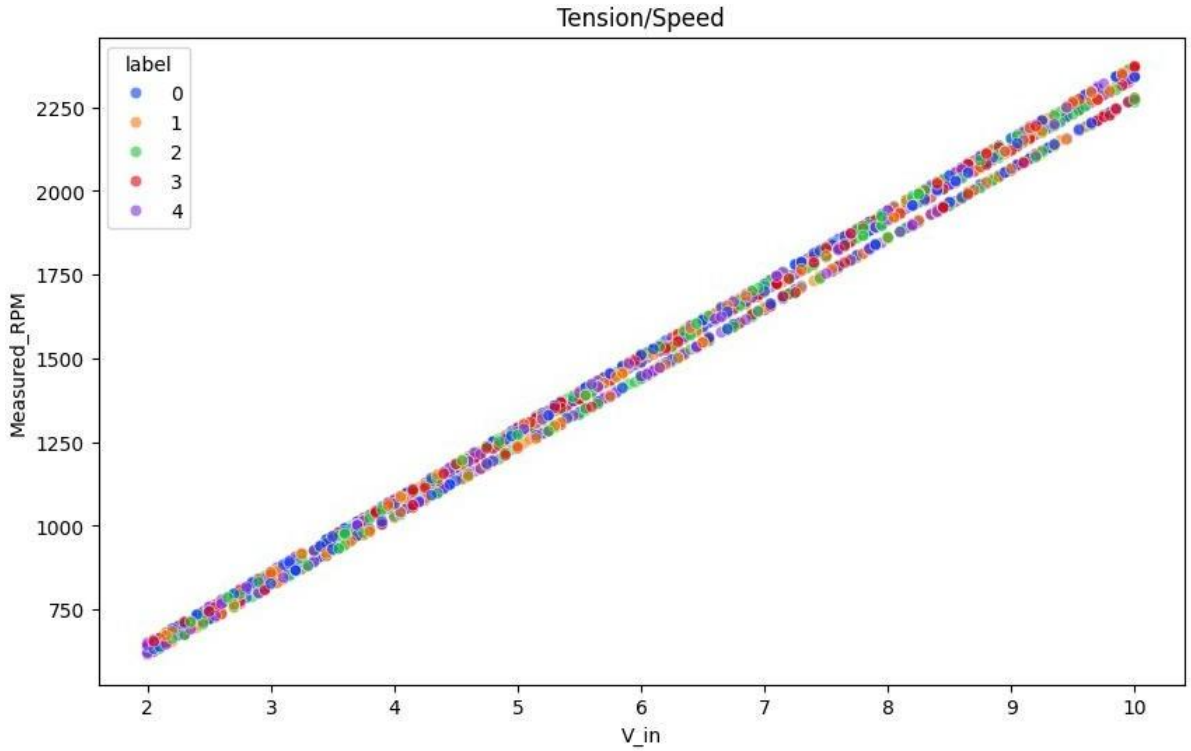
We now turn our attention specifically to the V_in column.
This step helps us **understand how the voltage values are distributed** across each dataset and identify which voltage ranges are overrepresented or underrepresented.

For intermediate voltages (2.05 V to 10 V), we will take a **user-chosen number of samples per voltage value**.

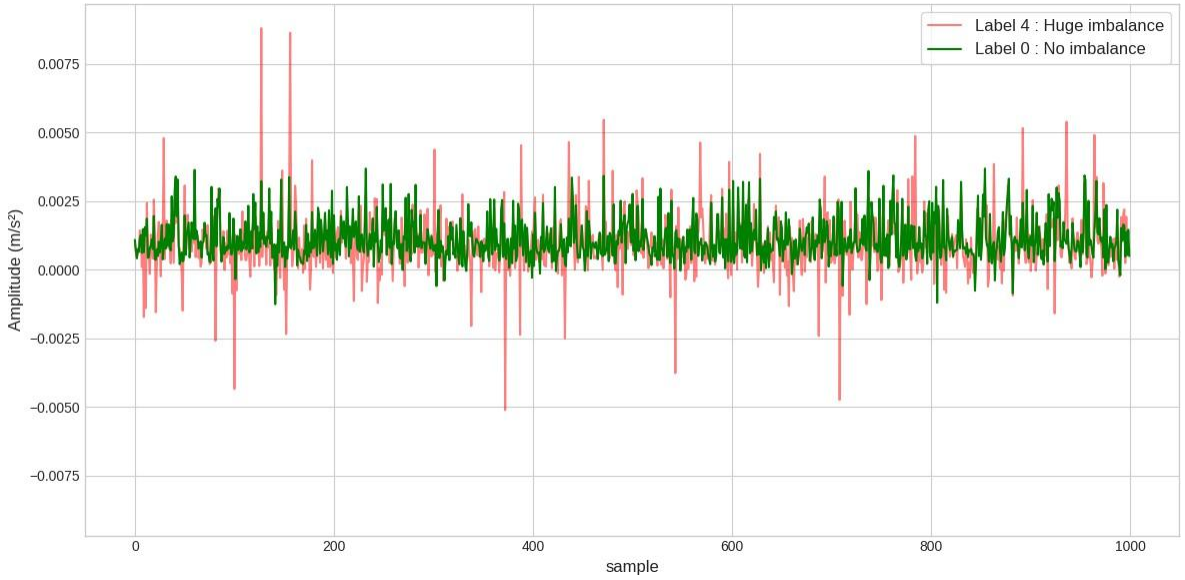
For 0 V and 2 V, which have very different numbers of samples, we keep the rows **in proportion to their original presence** in the dataset to preserve the initial distribution. It's also a way to avoid class imbalances.

3.1 Data Visualization

After merging the 5 datasets I started the data visualization. A simple way to see if the data are usable is to compare to physical phenomena that we can verify in real life. Here the speed of the motor increases accordingly with the tension, independently of the imbalance, which is expected.

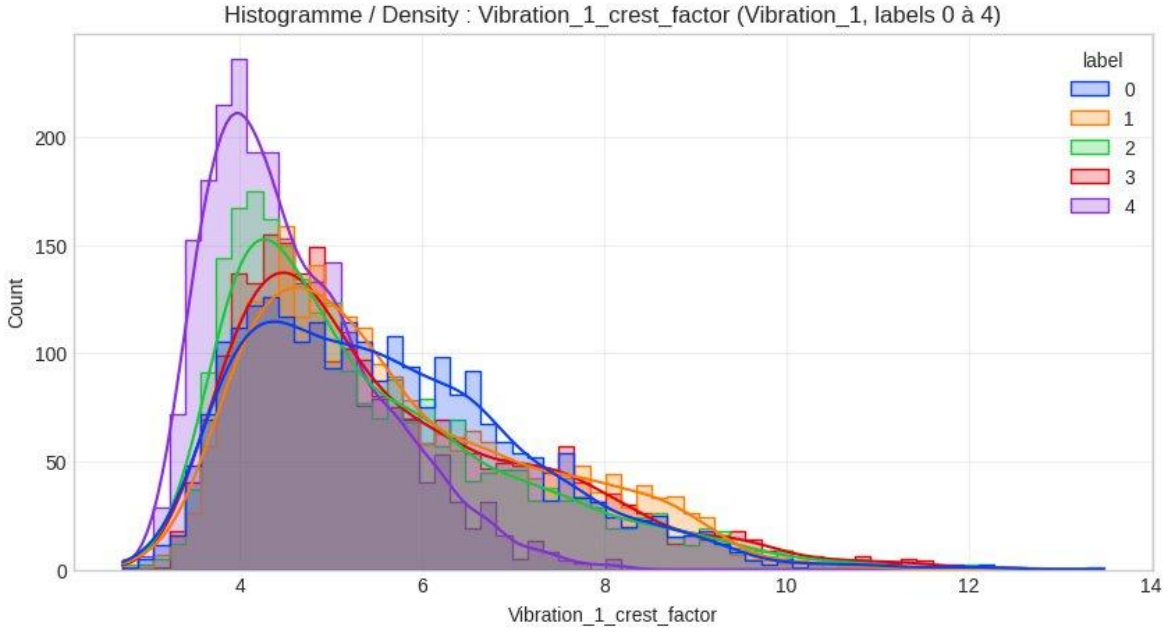


Generally, we might expect a vibration to be represented as a wave by a sinusoidal signal and then compare the sane signal with the imbalance one. However, due to noise, assembly inaccuracies, and measurement inaccuracies, this is not always possible.



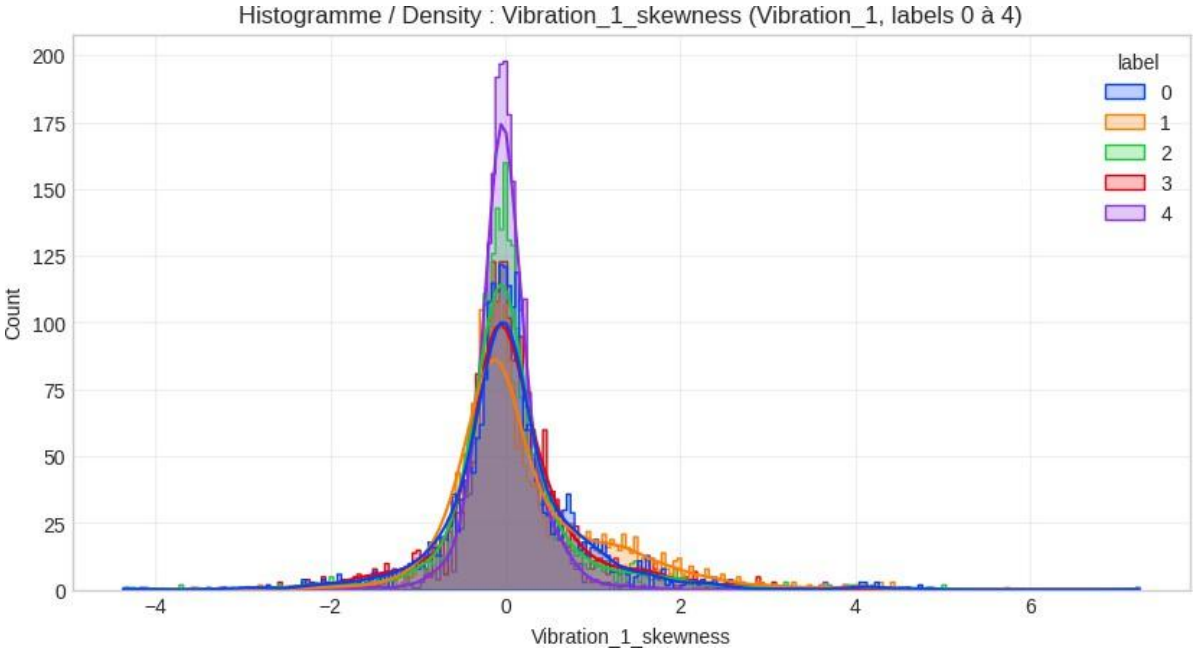
However, important information can still be obtained by working with the mean, or by creating other features such as skewness, kurtosis, or the crest factor. While **skewness** focuses on the spread of normal distribution, **kurtosis** focuses more on the height. Both tell us how peaked or flat our normal (or normal-like) distribution is.

We create new features called “skewness”, “kurtosis” and “crest_factor” in order to get more valuable information



"crest_factor" measures the signal’s impulsivity (e.g shock presence). One might think that higher imbalance implies more shock, implies higher “crest_factor” but it's not true.
In fact, it might cause a centrifugal force which tends to make the signal smoother.

Either way, it's a great feature to distinguish between label 4 and the other.



Skewness is the asymmetry of the signal, and this graph confirms the latter as label 4 is more centered in 0 than the other.

With these new temporal features, the difference between label 4 and label 0 becomes clearly apparent, as the signals in label 4 exhibit significantly more extreme and characteristic values.

In contrast, the separation between labels 1, 2, and 3 remains more subtle: their distributions of kurtosis, skewness, and crest factor overlap more, demonstrating that these temporal features alone are still insufficient to accurately distinguish these three intermediate states.

By training an XGBoost model solely on time-domain data, we observe that performance —particularly the F1 score — quickly reaches a plateau around 0.7.
Result: **inevitable confusion** between classes 1, 2 and 3.

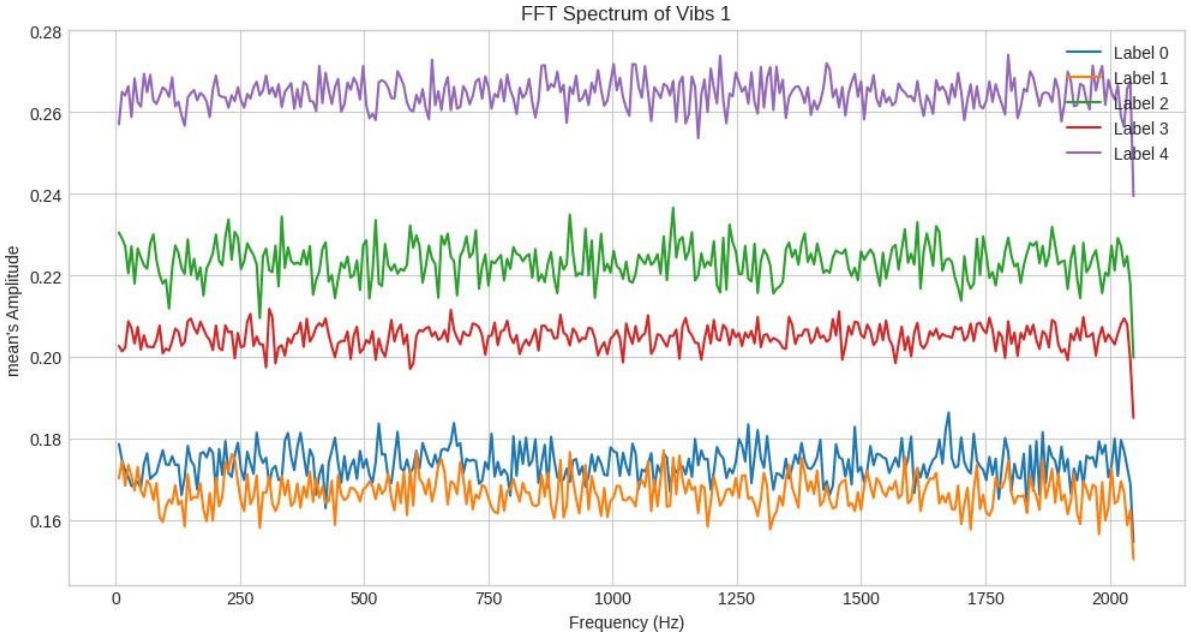
To overcome this limit, it therefore becomes relevant to move to the frequency domain, via FFT and spectral transformations, in order to look for structural differences in the frequency content of the signal and thus better separate labels 1, 2 and 3.

The FFT is a fast algorithm that converts a signal from the time domain to the frequency domain.

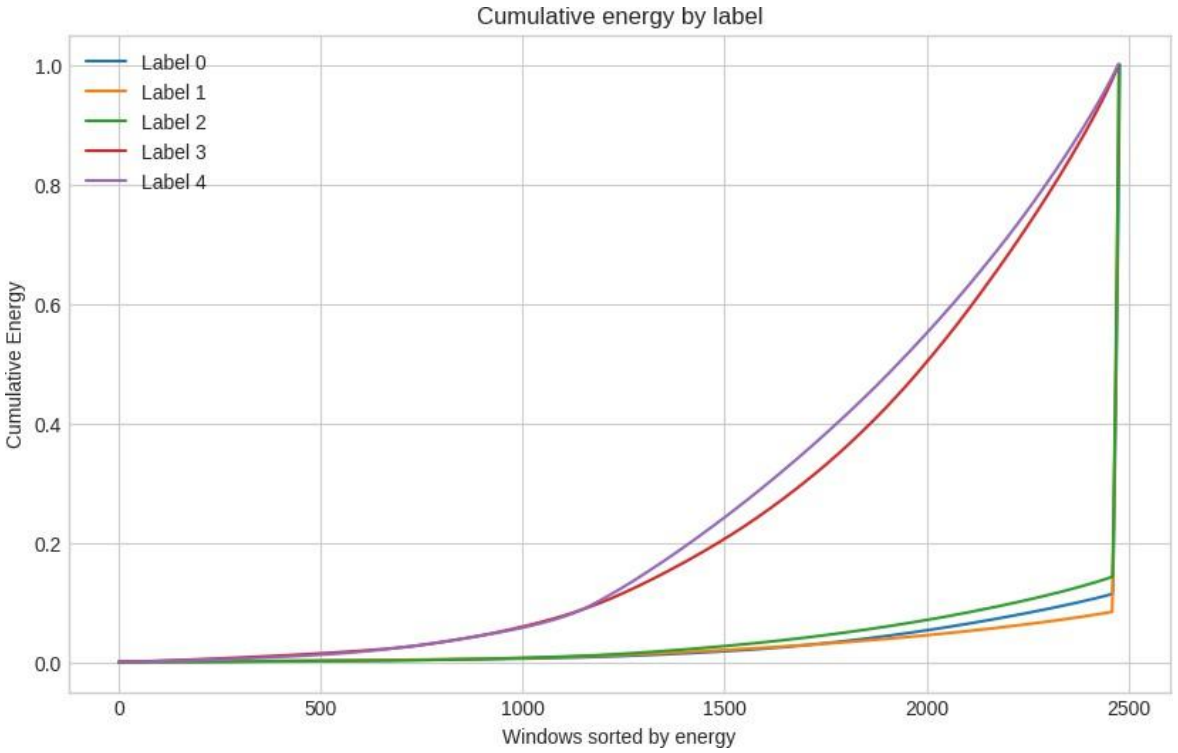
- It decomposes a complex signal into a sum of simple sinusoidal waves.
- It reveals invisible patterns in the raw signal like high oscillation and spectrum frequency pics.

- Window Size: High ==> we know what but not when
  Low ==> we know when but not why

==> optimal empirical value N=650



Thanks to the shift to the frequency domain, the analysis of the average amplitude spectrum as a function of frequency finally reveals a much clearer separation between the different levels of defect.



Our analysis shows that:

- Temporal features clearly distinguish between extreme cases: label 0 (healthy) and label 4 (major defect).
- Frequency features separate more clearly labels 1, 2, and 3, which remain too close in the temporal domain.

The two domains are therefore complementary. We will train our model on both temporal and frequency features.

4 Models Description & Limitation

4.1 Logistic Regression (raw data)

It serves as the baseline. Trained on raw temporal data, its primary limitation is its linearity. It makes it unable to model non-linear relationships between features.

4.2 Random Forest (raw data)

A non-linear set of multiple decision trees, using class weights to correct the inherent imbalance in fault state data (imbalance). Its limitation is the lack of feature engineering. Without statistical indicators like Skewness or Kurtosis, noisy information makes it unusable.

4.3 XGBoost (with only temporal features including Skewness & Kurtosis)

Gradient-boosted trees optimized for multi-class classification. This model exploits advanced temporal statistics such as skewness and kurtosis, which are key indicators of shocks or wear in vibration signals. Its limitation is that it remains blind to the frequency domain, missing specific periodicities related to rotating defects.

4.4 XGBoost (with temporal and FFT features) and GridSearch for hyperparameter optimization

It combines time-domain characteristics (including skewness and kurtosis) with frequency-domain characteristics extracted from the Fast Fourier Transform (FFT). The FFT analysis is essential because it isolates the frequencies of interest related to machine defects. The model undergoes an exhaustive grid search to determine the optimal combination of its hyperparameters to maximize performance.

Features used:

FEATURE	DOMAIN	DESCRIPTION
RMS	Temporal	Represents the average power of the vibration signal.
PEAK-TO-PEAK	Temporal	The difference between the maximum positive and minimum negative amplitudes. Indicates the maximum displacement or shock severity.
CREST-FACTOR	Temporal	
SKEWNESS	Temporal	Measures the asymmetry of the signal distribution.
KURTOSIS	Temporal	Measures the tail-heaviness of the signal.
RPM-MEAN	Operational	Represents the machine's rotational speed. Vibration frequencies are directly proportional to the running speed.
FREQ-PIC	Frequency	The specific frequency where the highest amplitude peak occurs in the spectrum
AMP-PIC	Frequency	The magnitude/height of the dominant frequency peak

ENERGY_0_50	Frequency	The sum of spectral energy contained in the low-frequency range. Often associated with unbalance or misalignment.
ENERGY_50_100	Frequency	The sum of spectral energy contained in the medium-frequency range
ENERGY_100_200	Frequency	The sum of spectral energy contained in the high-frequency range. Often associated with harmonics defects.

5 Obstacles and Solutions

5.1 Memory Limitations with Full Dataset

The merged final dataset exceeded 5 GB with a maximum of **131 952 635 lines**. To overcome this problem, I used a user-chosen value for the number of values per tension's value to consider. I also preferred using Google Colab instead of a Jupyter Notebook because of RAM issue.

5.2 Frequency Domain

The transition to frequency domain required to choose a "window size" (N). Normally the window size depends on the sampling rate. $T = \frac{N}{Fs}$. But I tested different values of N manually and found that the best N = 650.

5.3 Time calculations

For Random Forest and the regression, the computational time was absurdly long for the number of lines in the final dataset.

5.4 Overfitting and optimization

To prevent overfitting, we use the following hyperparameters in a GridSearch:

- max_depth [7, 9, 11]: is the maximum depth of each tree. Limiting the depth is the first step against overfitting
- learning_rate [0.05, 0.1]: Lower values reduce the risk of overfitting but require more trees to converge.
- reg_alpha [0.001, 0.01]
- n_estimators[200, 400]: these values were chosen to balance with the learning rates, ensuring sufficient iterations

6 Results and Comparison

Model	Accuracy	Weighted F1	Notes
Logistic Regression	0.3457%	0.3566%	Trash as expected
Random Forest	0.5273	0.5514%	Disappointing
XGBoost	0.7546%	0.7621%	Finally, some results
XGBoost + FFT + GridSearch	0.9467%	0.9467%	Elon Musk is jealous

Table 1: Comparison of model performance on test set.

	precision	recall	f1-score	support
0	0.9756	0.9677	0.9717	496
1	0.9677	0.9677	0.9677	496
2	0.8962	0.9071	0.9016	495
3	0.9059	0.8949	0.9004	495
4	0.9880	0.9960	0.9920	495
accuracy			0.9467	2477
macro avg	0.9467	0.9467	0.9467	2477
weighted avg	0.9467	0.9467	0.9467	2477

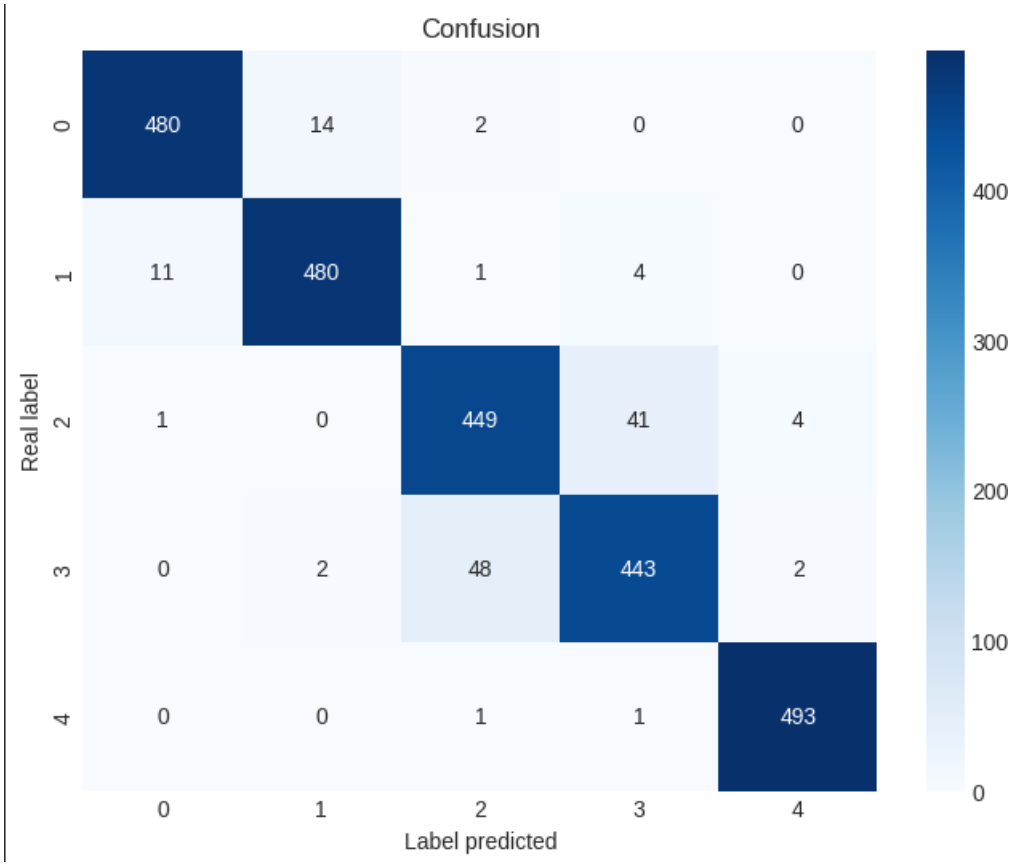


Figure 3: Confusion matrix

7 Conclusion

This project successfully developed a machine learning pipeline capable of classifying rotor imbalance levels (from 0 to 4) using vibration sensor data. By leveraging signal processing and predictive modeling, we established a robust workflow for preventive maintenance, designed to detect defects before they cause critical failures. This solution is directly applicable to the "SpectraV2" drone propeller test bench of the LeoFly association.

Possible Improvements
To further enhance the pipeline's capabilities and robustness, the following development could be made:

- Advanced Signal Processing:
  - Cepstrum Analysis: Implementing Cepstrum analysis would help better isolate defect harmonics within the noise, revealing periodicities that standard FFT might miss.

  - Short-Time Fourier Transform (STFT): Using STFT with sliding windows would allow for the analysis of frequency changes over time, providing a dynamic view of the signal rather than a static spectral snapshot.

- Deep Learning: Transitioning to Deep Learning models (such as ANN or CNN) could automate complex feature extraction directly from raw data or spectrograms, potentially uncovering deeper patterns.

8 References

- Dataset : Vibration Analysis on Rotating Shaft
- Analyse fréquentielle d'un signal par transformée de Fourier - Les fiches CPGE
- Comment entraîner une Forêt aléatoire (Random Forest) avec scikit-learn | LabEx
- Comment développer votre premier modèle XGBoost en Python
- GridSearchCV — scikit-learn 1.7.2 documentation

GitHub Repository