Multi-Layer Perceptrons for Predictive Maintenance

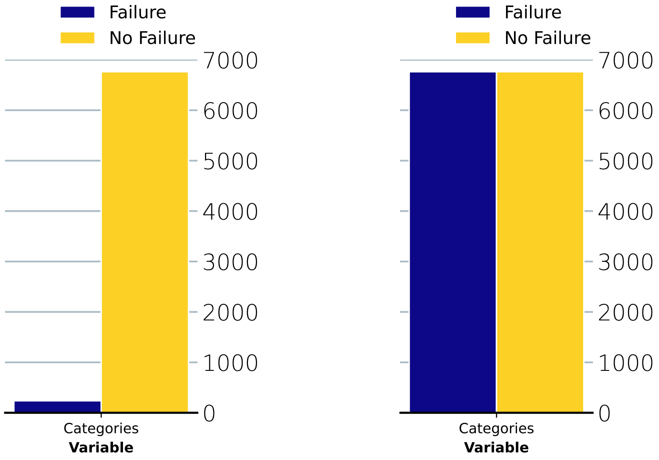
Khalid Kadri, Alexander Heckmann

I. Introduction

In this paper, we compare Matlab and PyTorch implementations of multi-layer perceptrons for predictive maintenance using binary classification. Predictive maintenance refers to data analysis and machine learning techniques to predict when maintenance is needed to prevent equipment failure.

II. Preprocessing

1) **Initial Data Analysis**: Assessing the distribution of predictors and check the target variable for imbalance.

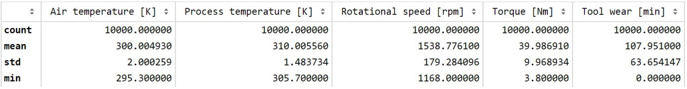


(a) Before oversampling

(b) After oversampling

Figure 1: Distribution of target variable in the training set

2) **Feature Selection**: Feature selection of relevant predictors based on domain knowledge.
3) **Encoding**: One-hot encoding for categorical data.
4) **Splitting**: Splitting data with a 70/30 ratio.
5) **Oversampling**: Oversampling with SMOTE-NC [1], [2] to balance the target variable. Undersampling was not feasible due to the massive imbalance of 237 records of class 0 vs 9661 records of the other (see figure 1).
6) **Normalization**: Scaling numerical features, preventing issues of sensitivity to different scales (see figure 2).



	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	300.004930	310.005560	1538.776100	39.986910	107.951000
std	2.000259	1.483734	179.284096	9.968934	63.654147
min	295.300000	305.700000	1168.000000	3.800000	0.000000

Figure 2: Difference in scale of numerical predictors

III. Model Performance

The results demonstrate that the best performance on our task was achieved using a model with 2 hidden neurons, 50 epochs, and a learning rate of 0.3. The best results were achieved with ReLU for the hidden layer and sigmoid for the output layer. This Matlab model architecture achieved the highest accuracy on the test set, outperforming all other models tested, with an accuracy score of 1. However, the same architecture resulted in an accuracy score of 0.966 in Python.

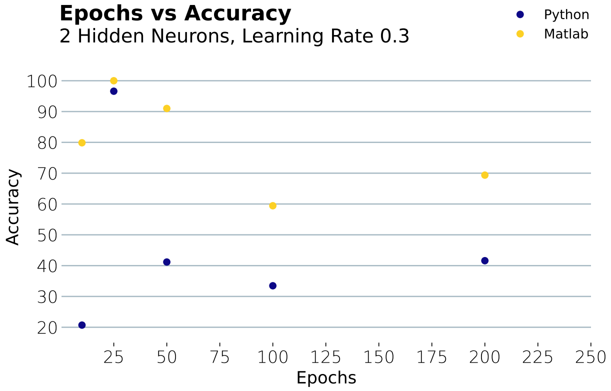


Figure 3: Change in accuracy for different learning cycles

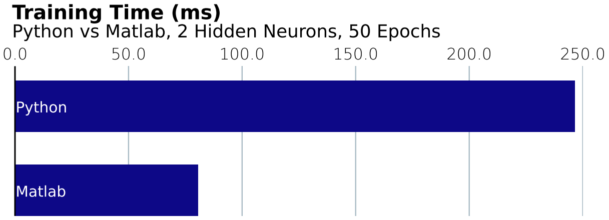


Figure 4: Average training time for best model (milliseconds)

Nevertheless, the Matlab model showed a considerable advantage in training speed, completing the training process in 80.64 milliseconds on average, while the Python model took 246.42 milliseconds on average for the same configuration.

References

[1] A. Buabeng, A. Simons, N. K. Frempong and Y. Y. Ziggah, “A novel hybrid predictive maintenance model based on clustering, smote and multi-layer perceptron neural network optimised with grey wolf algorithm”, *SN Applied Sciences*, vol. 3, no. 5, p. 593, 2021.

[2] A. Chazhoor, Y. Mounika, M. V. R. Sarobin, M. Sanjana and R. Yasashvini, “Predictive maintenance using machine learning based classification models”, in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 954, 2020, p. 012 001.