IEM 4103 Honors Contract

Spring 2025

Ethan O'Connor

Abstract

Estimating process health and modeling time-to-failure is a key concern in reliability analysis. Predictive Maintenance (PdM) is a technique that revolutionizes maintenance and reliability protocols by leveraging machine learning tools to predict when failures may occur. This study aims to apply several of the common methods described by Carvalho, et. al. in "*A systematic literature review of machine learning methods applied to predictive maintenance"* [1] to an online dataset to predict machine failure based on daily sensor readings.

Keywords: *Predictive Maintenance, Machine Learning, Support Vector Machine, Random Forest Classifier, K Neighbors Clustering*

I. Introduction

Reliability analysis, as studied in the IEM 4103 course, uses a variety of statistical tools to assess the health and effectiveness of a product or process. Methods employed in this course include modeling the mean time-to-failure, estimating hazard functions, accelerated life testing, and more. In the manufacturing environment, these reliability techniques can be applied to the production process, analyzing machines and components to create a maintenance plan that ensures the process is reliably producing high-quality products.

Traditional maintenance policies include Run-to-Failure (RtF), where corrective maintenance is applied only when a machine fails, and Preventive Maintenance (PvM), where maintenance is scheduled periodically to anticipate failure. Predictive Maintenance (PdM) revolutionizes these protocols by leveraging advanced machine learning tools to accurately predict when failures may occur. This policy has two key advantages:

1) Process downtime is reduced, since machines are not running to an unexpected point of failure (as in RtF)
2) Unnecessary scheduled maintenance (as in PvM) is reduced, avoiding early life failure (i.e., "infant mortality") that may result from improper maintenance

With these key benefits, it was demonstrated by Susto, et. al. [2] and the U.S. Department of Energy [3] that the operating costs under a PdM approach are generally much lower than operating costs under other maintenance policies, as unexpected downtime and unnecessary maintenance are reduced.

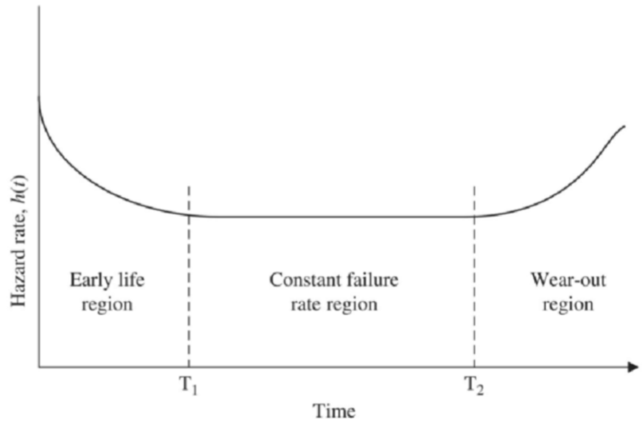

Figure 1: System failure typically follows a bathtub curve; by eliminating unnecessary maintenance, we reduce the risk of early life failure [4]

To adopt a PdM policy, there are some technological requirements. A large amount of historical data is necessary to train machine learning models for failure prediction. This data can be obtained with a variety of sensors installed on manufacturing machinery that are able to monitor key metrics such as temperature, vibration, noise level, and more. With these sensors constantly measuring key metrics for the machine or process, it is important to have an adequate information system in place to store this data for future retrieval. If these prerequisites can be met, though, the increasingly popular PdM policy has the potential to transform process reliability and maintenance for any manufacturing operation.

II. Dataset

The dataset used in this study was found on Kaggle.com [5]. It includes 124,494 rows of data with 12 columns: date, device, failure (binary), and metric 1 through metric 9. The "date" column spans from January of 2015 to October of 2015, indicating this data was collected or simulated over 10 months. Unfortunately, the metrics included in the dataset are not described, and are labeled generically. However, the available data is still sufficient to train a variety of machine learning techniques for predictive maintenance. Furthermore, several Python and R scripts are published with this dataset that provide examples for data cleaning or analysis.

III. Methodology

The implementation of predictive maintenance techniques was accomplished in Python, primarily using the Pandas and Scikit-Learn packages. The general methodology included the following key steps: data cleaning, exploratory analysis, data balancing, model fitting, and model evaluation. These will be described individually.

<u>Data Cleaning</u>

While this dataset is well polished and designed for ease-of-use, some data cleaning was required after importing the .csv file. The "date" column was converted from a string object to a M/D/Y format, and "metric8" was dropped, as it was an exact duplicate of "metric7". Furthermore, it was determined that 7 unique types of machines are included in the dataset under the "device" column. Data for machines belonging to the family "S1F0" were the most abundant, with more than 33,000 rows of data available. Since different machine families may follow different failure patterns, this study only examines data from machine family "S1F0"; all other types of machines present in the dataset were excluded.

Exploratory Analysis

After the data was imported and cleaned, exploratory analysis was performed. Basic statistics were calculated for each column to determine their average, standard deviation, minimum value, maximum value, and more. Next, the ratio of the number of datapoints denoting failures to the number of datapoints denoting non-failures was examined. Datasets that model machine or process failure are generally prone to a heavy imbalance, as the number of days that include failures should be far less than the number of days that constitute ordinary process functions. This dataset was no exception: 36 failures were reported, while 33,133 non-failures were reported. This issue will be tackled in the next subsection, data balancing.

Finally, to gain more insight into the general pattern of failures in the dataset, a line graph was constructed. This graph showed the number of failures across the dates included in the dataset, and is provided in Figure 2.

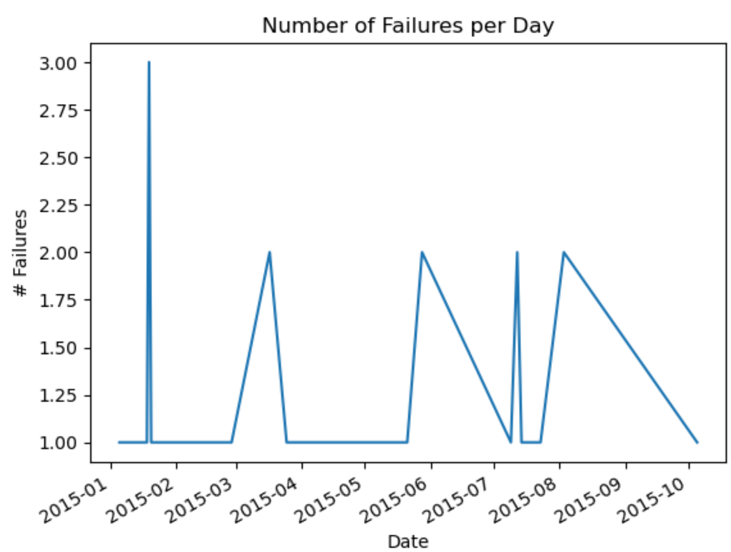

Figure 2: Graph visualizing the number of failures for machine family "S1F0" per day

Data Balancing

In order to correct the aforementioned imbalance present in the dataset, the Synthetic Minority Over-sampling Technique (SMOTE) was employed. This algorithm generates synthetic datapoints in the minority class by identifying a point's nearest neighbors and interpolating between the selected samples. This technique was applied using the Python "imblearn" package, which contains built in over-sampling functions. With this, we were able to balance the dataset, nearly doubling the number of rows. Double checking the outcome, it was confirmed that the balanced dataset contained 33,133 failures and 33,133 non-failures.

<u>Model Fitting</u>

This study sought to apply one or more of the predictive maintenance techniques described by Carvalho, et. al. in "*A systematic literature review of machine learning methods applied to predictive maintenance"* [1]. In this comprehensive literature review, Support Vector Machines (in our case, a support vector classifier, or SVC), Random Forest Classifiers, and K-means algorithms are described as methods frequently applied in PdM studies. Carvalho, et. al. cites numerous studies to support this claim, describing various scenarios in which each technique was effective. Thus, these three methods were applied to the Kaggle dataset for this study.

After the dataset was split into training and testing sets, these models were fit using built-in functions in the Scikit-Learn package. The only method that required additional parameters was the KNeighborsClassifier technique, for which a number of nearest neighbors, *k*, must be specified. To determine the optimal value of *k*, a range of values was tested, and the prediction accuracy of each was recorded. It was determined that *k*=4 produced the greatest prediction accuracy, as shown in Figure 3. After determining this optimal values, all models were fit to the training dataset, looking to predict the failure of the machine based on the values of metrics 1-9.

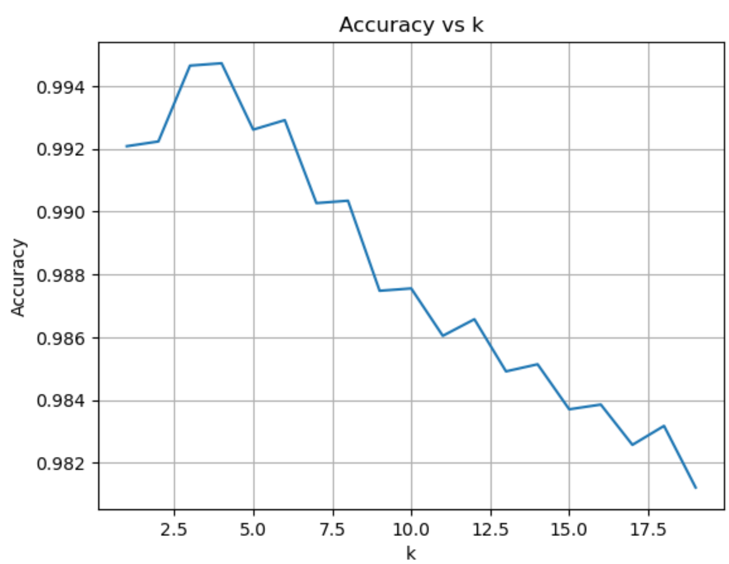

Figure 3: Accuracy vs, value of k in the KNeighborsClassifier method

<u>Model Evaluation</u>

The performance of all three models – SVC, Random Forest, and K Neighbors – was evaluated by their accuracy, precision, recall, and F1 scores. Built-in Scikit-Learn functions were used to compare the predictions of each classifier with the ground-truth test dataset. The results of these evaluations are shown in Figure 4.

method	accuracy	precision	recall	f1
SVC	0.916855	0.962264	0.865464	0.911301
Random Forest	0.921080	0.867952	0.990827	0.925328
K Neighbors	0.994719	0.990153	0.999236	0.994674

Figure 4: The performance metrics of each classification method

IV. Conclusions and Future Work

From the results presented in Section IV, it is clear that the K Neighbors Classifier technique outperformed other methods in each evaluation metric. For our dataset, a K Neighbors technique is ideal for evaluating the various metrics that were recorded and predicting whether or not a machine will fail. The algorithm can classify failure with an accuracy of 99.5%, which is an extremely strong performance.

In order to implement predictive maintenance with the determined prediction technique, the manufacturing facility would need to implement a real-time monitoring system that tracked metrics 1-9 from the dataset. The monitoring system would use these real-time metrics to perform a K Neighbors classification. From the case study, we expect the algorithm to be able to classify failure with very high accuracy. This would thus serve as an early-warning system, notifying maintenance personnel when the algorithm predicts an imminent failure. The machine could be shut down and repaired to avoid this catastrophic failure, saving the facility time and money with the PdM policy.

In terms of future work, this analysis could be performed for all machine types present in the dataset, not just the S1F0 family. Furthermore, regression models could be fit to examine the regression coefficients of each metric, allowing us to determine which metrics have the greatest impact on the failure of the process. These metrics could be monitored more closely than the rest. There are a wide variety of statistical techniques that can be applied to predict machine failure; experimenting with even more of these techniques could allow for superior failure prediction and the implementation of an effective PdM policy.

V. Works Cited

[1] Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024. https://doi.org/10.1016/j.cie.2019.106024

[2] Susto, G. A., McLoone, S., Pagano, D., Schirru, A., Pampuri, S., & Beghi, A. (2013). Prediction of integral type failures in semiconductor manufacturing through classification methods. *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, 1–4. https://doi.org/10.1109/ETFA.2013.6648127

[3] United States Department of Energy. Chapter 5: Types of Maintenance Programs. *O&M Best Practices Guide, Release 3.0*. https://www1.eere.energy.gov/femp/pdfs/OM_5.pdf

[4] Liu, Chenang. Lecture 11: Reliability & Hazard Functions. *IEM 4103 Quality Control & Reliability* Analysis. Provided on Canvas.

[5] *Predictive Maintenance Dataset*. (n.d.). Retrieved April 30, 2025, from https://www.kaggle.com/datasets/hiimanshuagarwal/predictive-maintenance-dataset