Best Machine Ever
CDS-303-001 Final Project Report

Modeling and Mitigating Machine Failures:

Techniques in Predictive Maintenance and Model Training

Group 4

Yara Yaghi, Muhammad Arfin, Shakir Azami,

Paula Schultz, Austin Yoo, Sheena Gandham

George Mason University

January 2025 - May 2025

Table of Contents

Modeling and Mitigating Machine Failures:

Techniques in Predictive Maintenance and Model Training

2. Abstract

The objective of this project was to investigate the possible application of supervised machine learning in preventing factory machine failures by detecting and preventing the most common causes of failure. Predictive maintenance can allow organizations to transition from reactive or time-based maintenance strategies to more proactive, data-driven systems. We trained classification models that predict failure using data from AI4I 2020 dataset, such as machine failure logs and sensor readings. New feature calculation and class imbalance and model fairness boosting were a few of the outstanding data preprocessing procedures. We've focused on classification models including Decision Tree, Random Forest, and XGBoost. XGBoost was also optimized for fewer false negatives with a final f1 test score of 96.5%, 1.2% deviation from our training set, and test recall of 99%, deviating only by 0.6% from the training set. SHapley Addictive exPlantations (SHAP) was used to interpret model predictions that indicated temperature, power, and tool wear as major failure drivers. These results are in alignment with industry experience and allow for decision prioritization of areas of interest. Through our research, we developed actionable recommendations that guided real-world maintenance decisions. From our project, machine learning is demonstrated to bridge the gap between raw data from machines and effective maintenance planning that enhances levels of safety, levels of efficiency, and costs of manufacturing.

3. Introduction

3.1 Background on Machine Failures and Maintenance Costs

Machine failures can cause many disruptions for businesses, specifically causing issues in manufacturing, a loss of production hours, unplanned downtimes, and raise significant safety concerns. These failures can drive up the costs of products, reduce profit margins, and also increase operational risks. Studies show the average cost of downtime due to machine failures is approximately $532,000 per hour. Some of the world's largest manufacturers can lose up to a trillion dollars a year just due to machine failures. Other than the repair and replacement costs businesses may have to bear, they can also face delayed delivery costs resulting in customer dissatisfaction. The reduction of machine failures is crucial for efficiency and the long term success of businesses.

3.2 Business Problem Statement

Through this project, our main aim was to figure out how we can reduce the frequency of factory machine failures by identifying and addressing the most impactful failure causes. Machine failures can result in a great financial loss, cause operational issues, and increase safety risks. It is important to understand the most impactful failures in order to help businesses optimize their maintenance strategies, control finances, and ensure consistent production hours. This can help make sure profitability is protected, machines are effective and reliable, and that the working environments are safe.

3.3 Project Objectives

In this project, we wanted to identify the most impactful types of machine failures in businesses. Our main objective was to be able to provide ways businesses can work towards reducing machine failures. This will help businesses towards minimizing their financial losses, having better quality machines, and ensuring optimized productivity. Investigating the most impactful failure causes through this project can hopefully provide insight into the main causes of machine failures and how businesses can reduce them for greater reliability.

4. Data and Assumptions

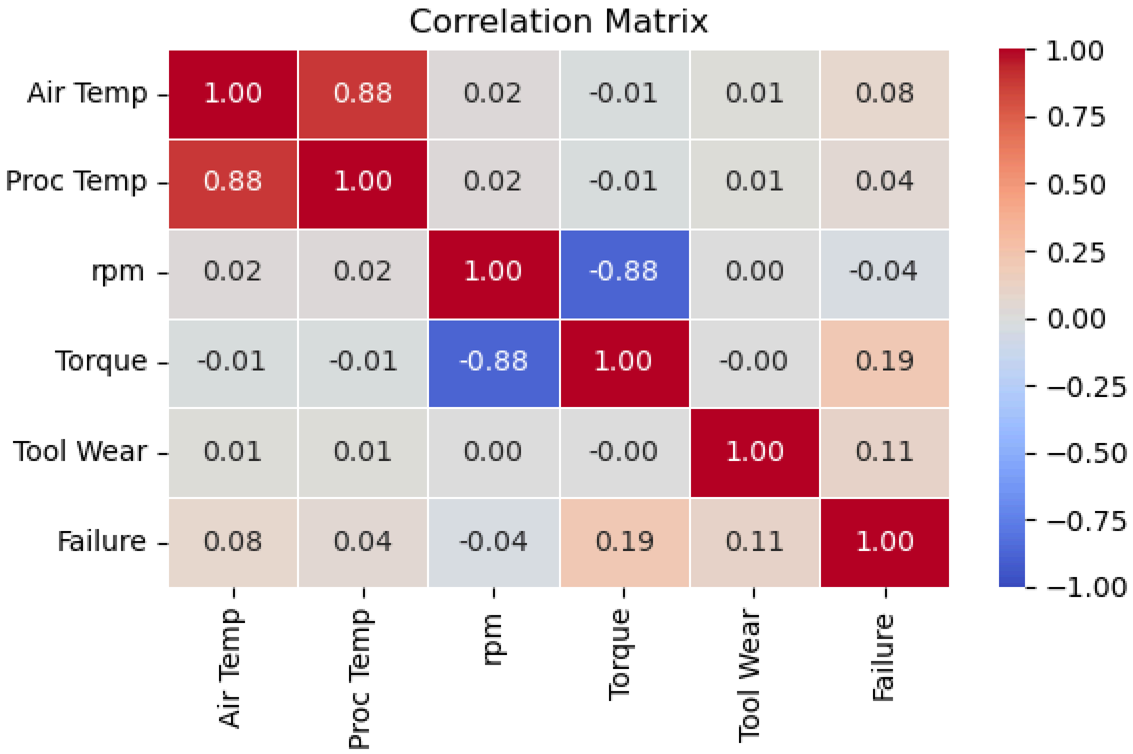
4.1 Dataset Overview

This dataset contains machine operation data including air temperature, process temperature, rotational speed, torque, and tool wear, along with binary indicators for machine failure types. There are unique identifiers for each record and labels for status of the machine.

4.2 Initial Assumptions and Pre-Cleaning Observations

We noticed there were no missing values. UID and Product ID were dropped as there wasn’t useful information, since we’re not trying to identify specific instances. The data was also synthetically generated, but is assumed to be an accurate representation of failure cases.

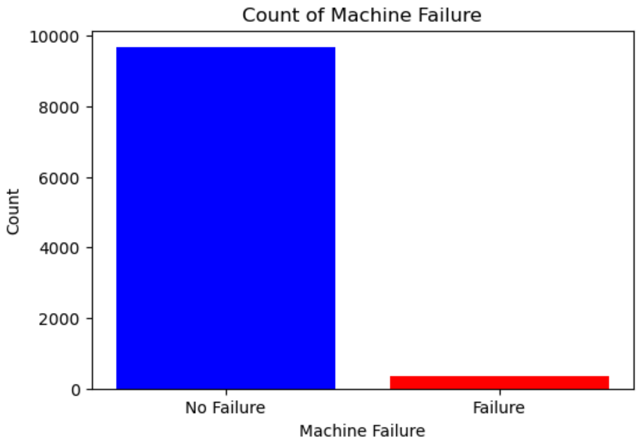
4.3 Data Cleaning, Feature Engineering, and Normalization

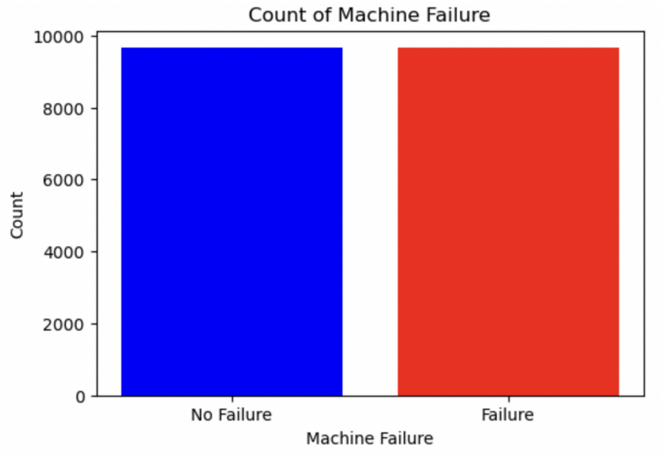


With a filtered correlation matrix, we could see some highly correlated relationships, and decided to merge Air Temperature and Process Temperature into temperature difference by subtracting Air Temperature from Process Temperature, along with Rotational Speed and Torque into Power by multiplying the two features together. They showed high relationships with each other, so reducing features would be helpful for improving model performance and preventing overfitting. Machine type/quality was merged into tool wear time by subtracting the additional minutes varying quality tools would provide: 5 for high, 3 for medium, 2 for low, in accordance with the information given. There were also values with widely varying ranges, so these features were normalized to prevent the scaling from causing an undue influence.

4.4 SMOTE and Class Balancing

Before applying SMOTE, the data was heavily imbalanced with many more ‘No Failure’ than ‘Failure’ results. This type of imbalance was a danger to the development of biased models that would struggle to accurately predict machine failure. We applied SMOTE(Synthetic Minority Over-sampling Technique) to balance the classes. Unlike traditional oversampling, which merely duplicates existing minority class samples, SMOTE generates new synthetic examples by interpolating between nearby existing samples. This approach reduces the risk of overfitting and increases the ability of the model to learn useful failure patterns. Undersampling was also avoided since it would likely have lost useful information. As can be observed from the data distribution plots, before SMOTE, the machine failures were vastly underrepresented. After SMOTE, the classes were evenly balanced, providing a robust and fair model.





5. Methodology and Ongoing Results

5.1 Model Selection, Initial Performance and Tuning Setup

Based on having a labeled target value that we generated where any or multiple types of failures were condensed per data point into a boolean feature for failure indication, we chose 3 models for our evaluation and testing: decision tree, random forest, and XGBoost.

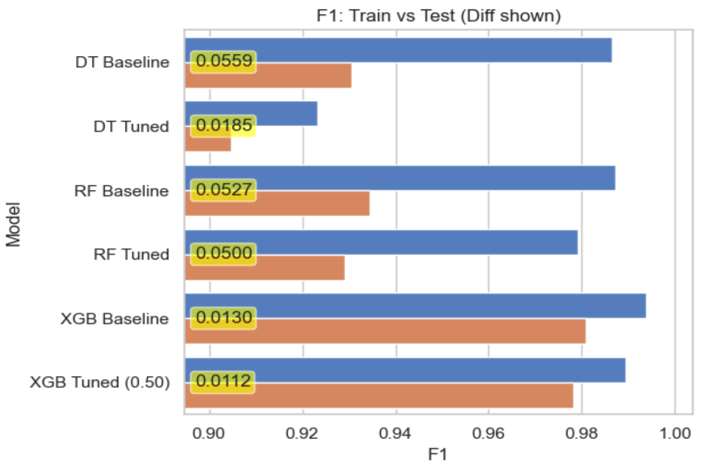
Cross validation was used consistently across all models using 5 fold split to ensure model generalization and robustness. All tuning and training was done with emphasis on preventing overfitting, focusing on minimizing the difference between test and training data, and reducing depth through hyperparameter tuning, regularization, and early stopping.

Model	CV/Tuning Strategy / Technique Used	Tuning results/ Notes
Decision tree	GridSearchCV: criterion: ['gini', 'entropy'] max_depth: [6, 8, 10] min_samples_split: [5, 10, 15] min_samples_leaf: [2, 4, 5, 7] ccp_alpha: [0.0, 0.001, 0.01]	Tuned: 'ccp_alpha': 0.0, 'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 5
Random forest	RandomizedSearchCV: n_estimators: 50-200 max_depth: 3-20 min_samples_split: 2-10 min_samples_leaf: 1-4	Tuned: 'max_depth': 18, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 160

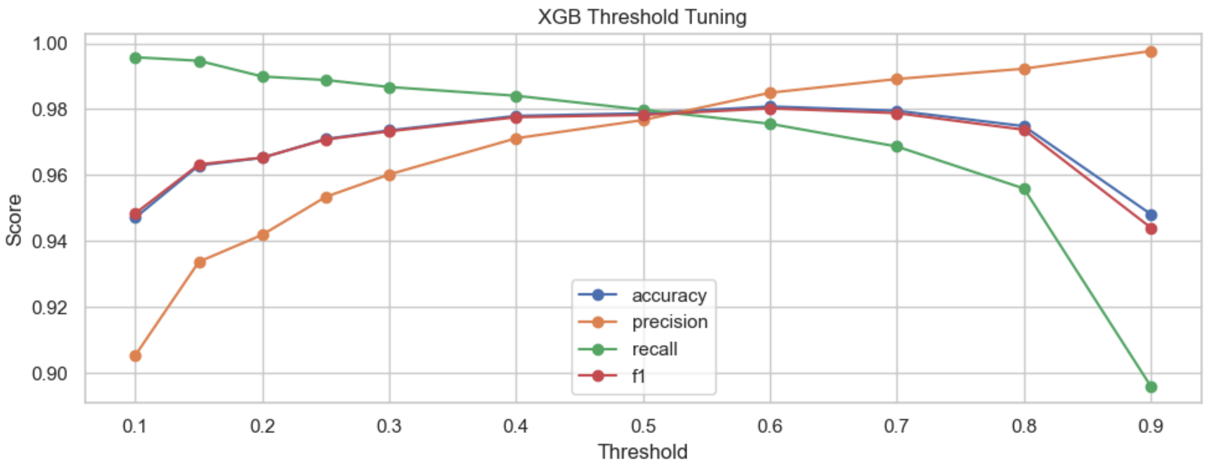
XGBoost	RandomizedSearchCV and RepeatedKFold: n_estimators: 100-300 max_depth: 4-10 learning_rate: 0.01-0.15 subsample/colsample_bytree: 0.7-1.0 gamma/reg_alpha/reg_lambda: 0-1 / 0-1 / 1-3 (Regularization)	GPU/CUDA optimized Tuned: 'subsample': 0.9, 'reg_lambda': 1.5, 'reg_alpha': 0.1, 'n_estimators': 200, 'max_depth': 10, 'learning_rate': 0.1, 'gamma': 0.5, 'colsample_bytree': 0.9

5.2 Results with Train vs Test Differences for DT, RF, XGBoost

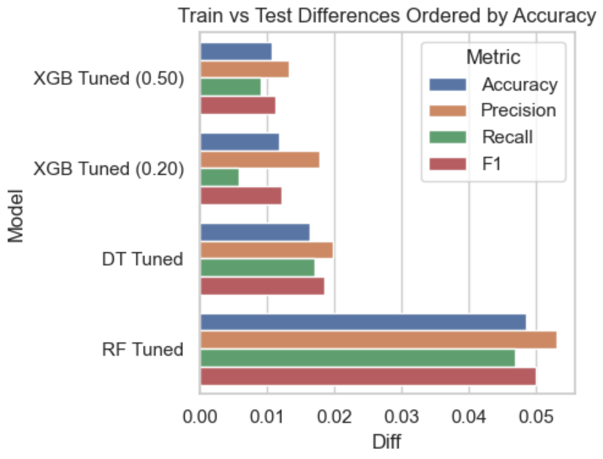
The models were tested in both baseline and tuned configurations. XGBoost is a clear winner here, with the XGBoost tuned version (at default threshold of 0.5) having a high f1 score of near 0.99, and a difference of .011 between training and test data.

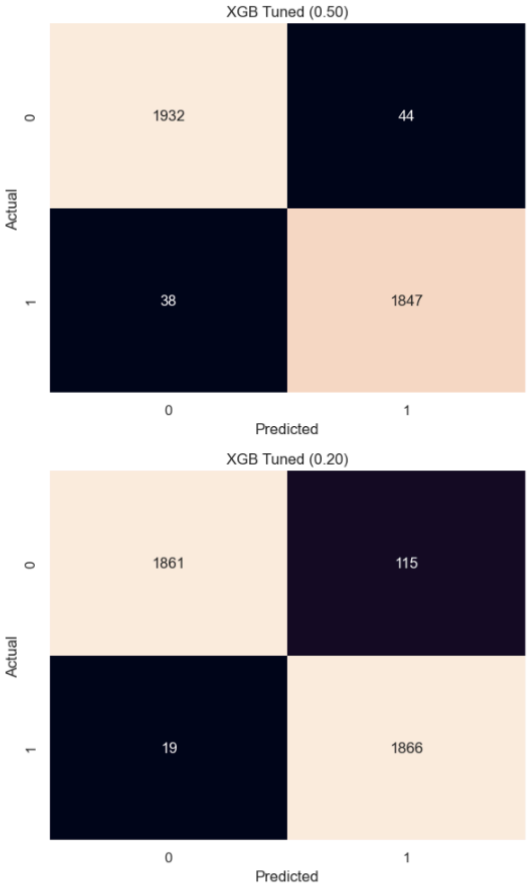


5.3 Recall Optimized Threshold Tuning for XGBoost



With Tuned XGBoost as a clear winner, the threshold was tuned to be recall optimized, since false negatives are considerably more expensive in context than false positives. With a threshold of 0.2, the recall optimized model shows a difference of less than 1% between test and train scores, a difference in f1 score is around 1.2%, far better than the other tuned models.





6. Final Tuned Model Results and Evaluation

6.1 Final Tuned Model Scores - Test vs Train

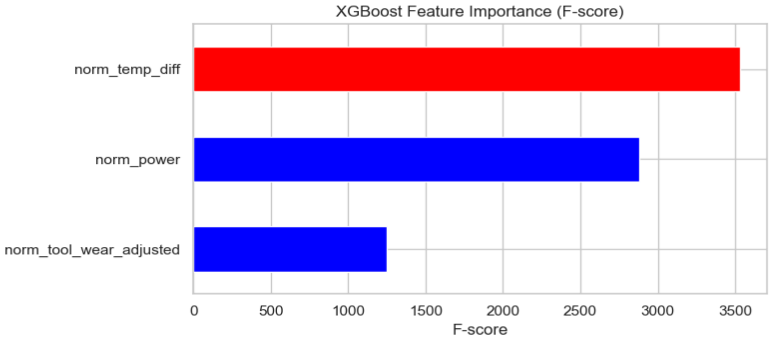
For XGBoost Tuned, with Recall focused 0.2 Threshold

Metric	Test	Train	Difference
Accuracy	0.965294	0.977051	0.011757
Precision	0.941949	0.959852	0.017904
Recall	0.98992	0.995752	0.005832
F1	0.965339	0.977473	0.012134

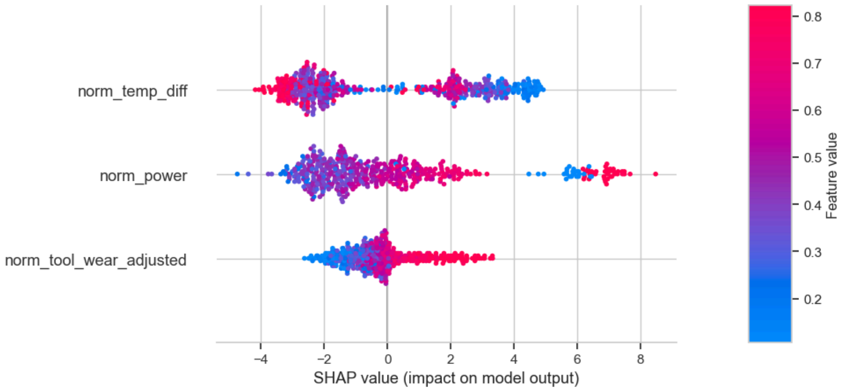
This shows incredibly good results, with as minimal overfitting as possible, though there's potential issues because of the synthetic nature of our initial data, as will be discussed.

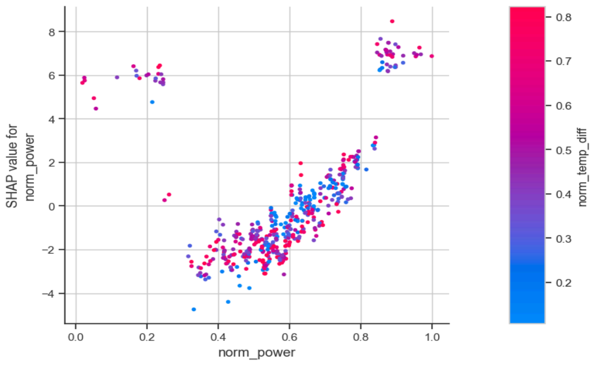
6.2 Feature Importance and SHAP Plots

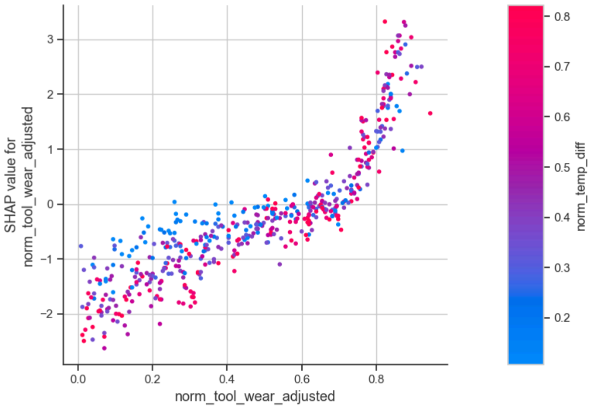
Using XGBoost xgb.plot.importance, which evaluates the relative contribution across all decision trees, we identify temperature difference as the most important contributing factor, followed by power, then tool wear.



Each point in the beeswarm plot shows an instance’s SHAP value on the x-axis (its impact on the prediction), is colored from blue (low) to red (high) by the feature’s normalized value, and is stacked by frequency.



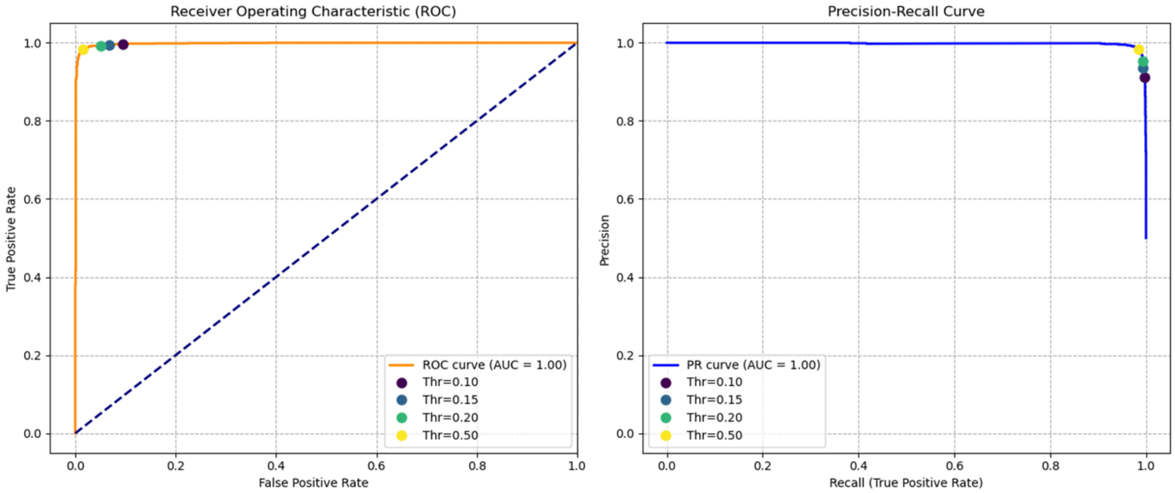




We can see from the SHAP beeswarm that a larger temperature difference is greatly correlated with fewer failures. In the tool wear and temperature difference SHAP plot, we can see that for lower tool wear, larger temperature difference also correlates to fewer failures. The power and temperature plot’s relative agnosticism and clear cutoffs suggest that power type failures may also be less influenced by temperature.

6.3 ROC/PR Curves for Threshold Tuned XGBoost

The model's performance was assessed using both the Receiver Operating Characteristic (ROC) Curve and the Precision-Recall (PR) Curve, as shown in the figure below.



AUC-ROC = ~1.00 and AUC-PR = ~1.00 demonstrate that the model is highly effective at distinguishing between positive and negative classes.The performance is consistent across different thresholds, which provides flexibility for operational use depending on the desired balance between precision and recall.

7. Discussion

7.1 Interpretation of Model Performance

The recall-tuned XGBoost, with a 0.2 threshold, achieved 96.5% accuracy, 94.2% precision, 99.0% recall (F_1 = 96.5%) with only ~1% train–test gap, indicating minimal overfitting and strong generalizability.

The recall-based threshold tuning actually altered the behavior of the model to detect as many real failures as possible - a critical necessity in predictive maintenance use cases, where not detecting a failure (false negative) is much more costly than a false alarm (false positive). The AUC-ROC and AUC-PR scores near 1.00, the model demonstrates high discriminatory power between failure and no-failure events even at different thresholds.

Feature importance and SHAP analysis revealed the most important feature as temperature difference, followed by power and tool wear. This makes sense and aligns well with domain knowledge: overheating, mechanical load, and degradation of tooling are common precursors to machine breakdowns.

Nonetheless, while the results are impressive, caution is warranted. Because the dataset was artificially created, there is definitely a risk that the performance of the model is overly optimistic relative to how the model would perform on noisy-real world data. In practice, unforeseen noise, missing sensor data, and unmodeled machine behaviour could reduce the model's effectiveness significantly.

In short, the model performed extremely well in the experimental setup, with high recall at controlled loss of precision and excellent generalization from training to testing, The findings show that XGBoost with engineered features and proper threshold tuning provides an excellent predictive maintenance tool when validated and transferred to reality, and if the synthetic data was well generated, we have a fantastic tool to use.

7.2 Insights for Real-World Application

Our findings have several key insights that can be directly applied to guide real-world maintenance practice. Conditions of operation, like temperature difference, power, and quality-adjusted tool wear were the most significant contributors to machine failure consistently, according to SHAP analysis. Factories can use these results to prioritize the most failure-prone conditions and components and enable more focused monitoring and priority setting. Instead of distributing maintenance activities across all machines, businesses can focus on the high-risk regions forecasted by the model. This enables better use of labour, time, and resources, especially in high-throughput production environments.

7.3 Comparison to Previous Studies

Our research is a continuation of and divergence from several recent predictive maintenance researches. In Stephen Matzka's "Explainable Artificial Intelligence for Predictive Maintenance Applications" (2020), traditional supervised models like Decision Trees and Random Forests were employed on the same AI4I 2020 dataset with less complex classification models and minimal feature engineering efforts. Our research, however, involved extensive feature engineering (e.g., creating Power and Quality Adjusted Tool Wear) and ensemble modeling to obtain optimal predictive performance and generalizability. Nicoló Vago et al. "Predicting Machine Failures from Multivariate Time-Series: An Industrial Case Study" (2024) was a work on time-series analysis using deep learning models like LSTM networks for recognizing sequential patterns to failures. Contrary to their dynamic modeling, we relied on static snapshot data and tree-based supervised models for real-time failure classification. In Mfundo Nkosi et al.'s "Causes and Impact of Human Error in Maintenance of Mechanical Systems" (2020), human factors like poor training leading to mechanical failure were emphasized. Our study differed by focusing solely on machine sensor data (i.e., rotational speed, temperature, torque) without variables involving human behavior. Finally, Devendra Yadav et al.'s "Predicting Machine Failures Using Machine Learning and Deep Learning Algorithms" (2024) compared machine learning models and deep learning models to predict failure and found that ensemble methods like Random Forest and XGBoost performed best. This aligns with our model selection, though we only employed tree-based models to maintain interpretability and within our control, whereas Yadav's research also involved artificial neural networks (ANNs).

7.4 Limitations and Areas for Improvement

Although the performance of our predictive maintenance models was good - especially recall and accuracy-wise - they do have certain serious limitations and scope for improvement that need to be highlighted.

To begin with, our dataset was created synthetically, and that leaves us with an inherent limitation. Artificial data will also be cleaner and more stable than the real data, providing a situation where models will be biased towards exhibiting unrealistically good performance on the test set, as our almost perfect ROC and PRC curves testified. It is a matter of model generalizability. The good-performing figures may be indicating a too “clean” or ideal dataset for model robustness. Later releases have to be validated and tested on real true machine failure traces before trying to drive them into real production environments.

Second, even with all forms of overfitting avoidance techniques used - cross-validation, hyperparameter tuning using GridSearchCV and RandomizedSearchCV, and ensemble techniques - the residual traces of artificially introduced artifacts persist. For example, all-penetrating close-to-equal AUC values and low variance among the folds may be a property of the dataset and not model stability.

Also, even though we stored the trained model as JSON so that we won’t have to retrain it, our pipeline currently still does not possess a flat-out optimized process of pre-processing and scaling new incoming data to ready them for use in real-time. This limits real-time capability and indicates that there is a need for ongoing extra engineering effort in order to render the system a plug-and-play system for use in real-time.

Finally, while SHAP analysis assisted us in identifying which features were most important to failure predictions, we only touched one tip of the iceberg in terms of interpreting and acting on this data. Additional SHAP visualization interpretation would provide greater transparency to plant operators and allow maintenance crews to have greater trust in and use model predictions more confidently.

7.5 Final Answer to Business Question

Our business issue was: "How can we reduce the number of factory machine failures by identifying and correcting the most important failure causes?"

With heavy data cleaning, feature engineering, SMOTE-based class balancing, model checking, and hyperparameter tuning, we built machine learning models—Decision Trees, Random Forests, and XGBoost—which have shown high predictive potential if we assume the synthetic AI4I 2020 dataset is representative enough of real manufacturing processes. While the artificially generated nature of the data creates some constraints on direct prediction, the models continue to provide sound information on which of the machine sensor features most closely correlates with failures, which helps in prioritizing real maintenance work and system upgrades.

Our derived features, i.e., Temperature Difference (process temperature vs. air temperature), Power (from Torque × Rotational Speed), and Quality-Adjusted Tool Wear, were selected to be easy to calculate from sensors available in most modern manufacturing systems. These inputs facilitate the construction of an input workflow that can easily be fulfilled and integrated into real-time systems. For example, torque and RPM can be multiplied to give power, or power can be measured and calibrated with small adjustments. Tool wear, already in general monitoring, can be further made more accurate with quality adjustments to forecast failures better.

This readily available sensor output can now be fed into a robust, well-generalized, and well-validated deployed model. In real-time settings, the model can monitor precursory signs, initiating predictive maintenance interventions ahead of failure. In addition, it can enable dynamic site-by-site maintenance planning, with periodic corrective updates through ongoing metric monitoring. Deployment may take place either at the plant level with local server solutions or, ideally, with a cloud-based solution where data from many sites may be pooled to optimize model updates and generalization.

Furthermore, integrating the model into existing factory management systems can be made relatively easy by defining a documented API that streamlines preprocessing (e.g., normalization and encoding) without requiring significant software overhauls. This would simplify adoption, and make predictive maintenance upgrades more affordable and accessible to manufacturers.

In practical terms, our findings lead to clear and actionable suggestions. Maintenance teams can choose interventions from initial warning signs like unusual temperature rises or power anomalies. Plant owners can plan specific system upgrades, like more efficient cooling systems, power input stabilization, and improved tooling schedules, from our models' analysis. Machine manufacturers may even design customized service packages or hardware upgrades, like temperature control units, smart tool wear sensors, or power conditioning modules to target causes of likely failures.

In conclusion, our models bridge the gap between raw machine data and effective decision-making. They show that machine failures can be significantly reduced by early detection, preventive maintenance strategies, adaptive system calibration, and intelligent product design. Our work gives a firm foundation for the deployment of predictive maintenance models that are not only accurate and interpretable but also feasible for real-world manufacturing implementation, and best of all able to deliver quantifiable improvements in operations that can save money.

References

[1] Stephan Matzka, 2020. Explainable Artificial Intelligence for Predictive Maintenance Applications. 2020 Third International Conference on Artificial Intelligence for Industries (AI4I), 69–74. https://ieeexplore.ieee.org/document/9253083

[2] Shubhangi Gurav, 2022. Review of Study of Effect of Misalignment on Rotating Shaft. ResearchGate. https://www.researchgate.net/publication/362517988_REVIEW_OF_STUDY_OF_EFFECT_OF_MISALIGNMENET_ON_ROTATING_SHAFT

[3] Mfundo Nkosi, P. M. Mashinini, 2020. Causes and Impact of Human Error in Maintenance of Mechanical Systems. ResearchGate. https://www.researchgate.net/publication/340402281_Causes_and_Impact_of_Human_Error_in_Maintenance_of_Mechanical_Systems

[4]Dave Westrom, 2020. Top Causes of Machine Failure and How to Prevent Them. MachineMetrics Blog. https://www.machinemetrics.com/blog/machine-failure

[5] Vortec, 2020. Electronic Equipment Failures: Cause, Effect and Resolution. Vortec. https://www.vortec.com/en-us/electronic-equipment-failures-cause-effect-and-resolution?srsltid=AfmBOooYEDxsqKB8BBjsP3hw1zcUcf-nTPLloBGZXnO6gnXZZnmGOIFk

[6] Graceport, 2020. Top 10 Electrical Failures by Cause. Graceport Blog. https://www.graceport.com/blog/top-10-electrical-failures-by-cause

[7] Steven Aliano, n.d. World's Largest Manufacturers Lose $1 Trillion/Year to Machine Failure. ISA Blog. https://blog.isa.org/worlds-largest-manufacturers-lose-1-trillion/year-to-machine-failure

[8] VIAR, n.d. The True Cost of Downtime from Human Error in Manufacturing. Rewo Blog. https://www.rewo.io/the-true-cost-of-downtime-from-human-error-in-manufacturing/

[9] Vecto, n.d. How Much Do Power Quality Disruptions Cost US Industry? Vecto. https://vectosystem.com/how-much-do-power-quality-disruptions-cost-us-industry/

[10] eWorkOrders, n.d. Dust Threat to Equipment. eWorkOrders. https://eworkorders.com/cmms-industry-articles-eworkorders/dust-threat/

[11] Marcellus, n.d. Equipment Malfunction | A Costly Business Challenge and How to Overcome It. Nonstop Group. https://reliability.thenonstopgroup.com/equipment-malfunction/

[12] Assembly Magazine, 2021. Equipment Failure is Costly for Manufacturers. Assembly Magazine. https://www.assemblymag.com/articles/96518-equipment-failure-is-costly-for-manufacturers#:~:text=%E2%80%9CWhen%20expensive%20production%20lines%20and,in%20almost%20all%20industrial%20sectors.%E2%80%9D

[13] Sensemore, 2024. Ensuring Machine Reliability: The Role of Predictive Maintenance. Sensemore. https://sensemore.io/ensuring-machine-reliability-the-role-of-predictive-maintenance/?srsltid=AfmBOorWTGILUkNlEbtbBzGPqp68jQav8N4DYDSTr109xwpIh4yAmO-Q

[14] Encord Blog, 2022. Introduction to Balanced and Imbalanced Datasets in Machine Learning. Imbalanced-learn.
https://encord.com/blog/an-introduction-to-balanced-and-imbalanced-datasets-in-machine-learning/
[15] Analytics Vidhya, 2020. Overcoming Class Imbalance Using SMOTE Techniques. Analytics Vidhya Blog.
https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/
[16] Megha Natarajan, 2023. Mastering Data Scaling Techniques: Visualizations and Insights. Medium.
https://medium.com/@megha.natarajan/mastering-data-scaling-techniques-visualizations-and-insights-a00b2cb422c2
[17] Ken Hoffman, 2020. Decision Tree Hyperparameters Explained. Medium.
https://ken-hoffman.medium.com/decision-tree-hyperparameters-explained-49158ee1268e
[18] GeeksforGeeks, 2025. Random Forest Hyperparameter Tuning in Python. GeeksforGeeks.
https://www.geeksforgeeks.org/random-forest-hyperparameter-tuning-in-python/
[19] GeeksforGeeks, 2021. Hyperparameters of Random Forest Classifier. GeeksforGeeks.
https://www.geeksforgeeks.org/hyperparameters-of-random-forest-classifier/
[20] Analytics Vidhya, 2025. XGBoost Parameters Tuning: A Complete Guide with Python Codes. Analytics Vidhya.
https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/
[21] Vincent Ciancio, Lazhar Homri, Jean-Yves Dantan, Ali Siadat, 2020. Towards prediction of machine failures: overview and first attempt on specific automotive industry application. IFAC-PapersOnLine, 53(3), 289–294.
https://www.sciencedirect.com/science/article/pii/S2405896320301932
[22] Devendra K. Yadav, Aditya Kaushik, Nidhi Yadav, 2024. Predicting machine failures using machine learning and deep learning algorithms. Engineering Reports, 3, 100029.
https://www.sciencedirect.com/science/article/pii/S2667344424000124
[23] Nicolò Oreste Pinciroli, Francesca Forbicini, Piero Fraternali, 2024. Predicting Maintenance Using Vibration Analysis: Machine Learning Approaches. Machines, 12(6), 357.
https://www.mdpi.com/2075-1702/12/6/357