Machine Failure Classification

Jefrey Bethel

1 Introduction

Industrial machinery failures are quite common and reducing associated costs and downtime is important. As such, it must be accurately predicted for continuous working. Industrial systems contain complex interactions between components and operating conditions. The problem is challenging because of the dynamic nature of the system involving many moving parts. Through systematic data analysis, feature engineering, model development, and optimization, the project aims to develop accurate and explainable predictive models.

An ensemble model combining multiple logistic regression models as base estimators was best suited for the classification problem. The developed AdaBoost ensemble model achieved over 97% accuracy with high precision in classifying machine failure. This demonstrates applying advanced ML strategies to industrial data can provide non-intuitive learnings on asset reliability factors. Developing high-fidelity failure classification models carries practical significance in minimizing maintenance costs in capital-intensive sectors. The techniques form a template for deploying machine learning to boost efficiency in industrial operations through predictive insights.

2 Methods/Case Study

The methodology followed a sequence of steps - data exploration and preprocessing, feature engineering, and model development, followed by optimization. The data consists of multivariate measurements from industrial machinery. There are features of numerical and categorical values. There are multiple input variables - air temperature, process temperature, rotational speed, torque, tool wear, etc. The target variable is a classification of failure. The data exploration analyzed statistical properties and handled missing values and redundancies. Scaling was done to make the features close to similar ranges. Many of the features had categorical values that needed to be transformed into numerical values. One hot encoding was applied to some features. The output target variable required label encoding. Correcting outliers is an important thing. There weren't significant levels of outliers present in the data.

A correlation analysis was done to identify various relationships. The analysis was found to be satisfactory. There was a correlation between some variables, but it didn't affect the training process significantly after regularization and boosting. The following is the correlation heatmap:

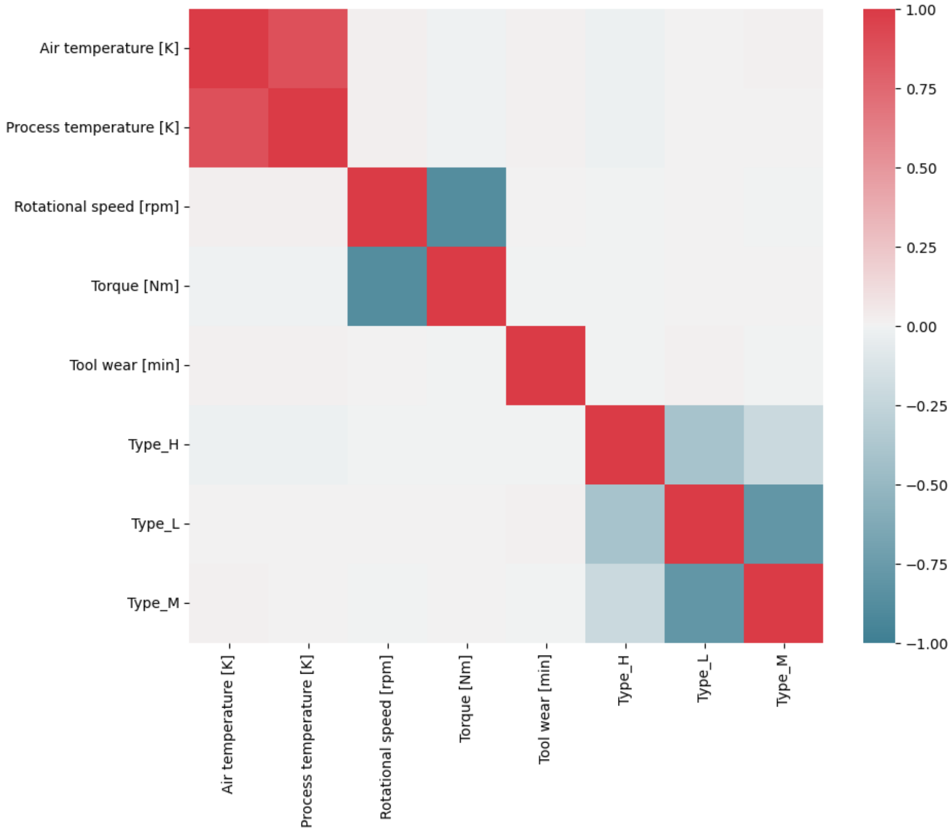

Figure 1: Correlation Heatmap

The heat map shows correlations between the input features used to train the machine learning model. Several moderate correlations are apparent, as indicated by the presence of off-diagonal colors.

After exploratory data analysis and preprocessing were done, the data was then modeled. Various machine-learning algorithms were utilized and applied to the data. Classifier algorithms such as SVM, Decision Tree, KNN, Random Forest Classifier, Naive Bayes, Logistic Regression, etc., were used to evaluate the performance of the data. In Table 1, it can be observed the accuracy and precision through the k-fold cross-validation evaluation of the models. The models that stood out were the logistic regression and SVC models. The logistic regression model came out with a better level of precision.

One of the ways to improve the results would be to utilize ensemble methods. An approach taken was to apply the voting classifier technique to the data. It is known to produce better predictive performance compared to a single model. A 5-fold cross-validation was applied.

Upon training, the voting classifier had the highest accuracy at 93%, which is an improvement

Model	Accuracy	Precision
Random Forest	0.89	0.61
Logistic Regression	0.97	0.80
SVC	0.93	0.68
Decision Tree	0.89	0.45
KNN	0.96	0.60
Naive Bayes	0.94	0.35
Gradient Boosting	0.88	0.62

Table 1: Model Accuracy and Precision

over the individual models except for logistic regression. However, its precision is lower than the individual logistic regression model's precision. This demonstrates the trade-off that can happen sometimes with voting classifiers - by combining multiple models, the accuracy can increase as more examples are correctly classified overall. But precision can potentially decrease if the voting introduces more false positives. Additional tuning showed similar performance across both accuracy and precision from the voting ensemble. The results show the potential of voting ensemble to improve accuracy over the individual models. However, upon applying stratified 2-fold cross-validation, the results improved by a lot as shown in Figure 2. Increasing further folds results in a degradation possibly from overfitting or underfitting each fold.

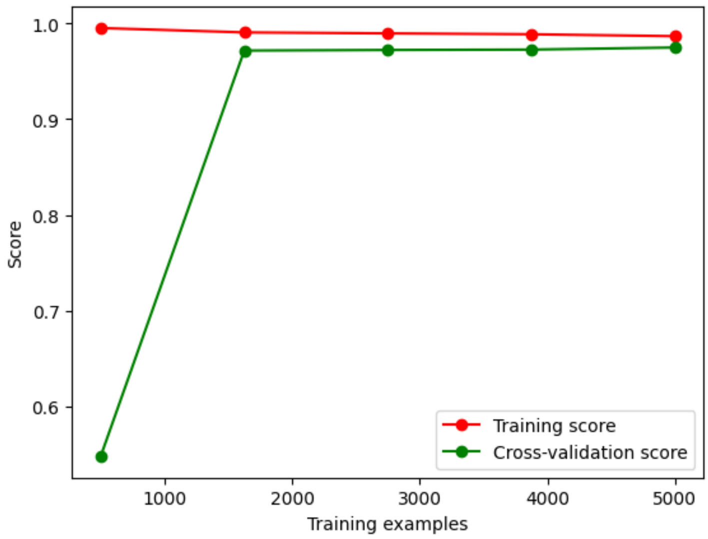

Figure 2: Training/Validation Score (Voting)

The accuracy improved reaching upto 97.5% and precision reaching upto 87%. Stratified cross-validation helped to ensure the distribution of labels is appropriately represented across the training and validation folds. This helps minimize overfitting during evaluation. The significant jump in both accuracy and precision metrics indicates the voting classifier was likely overfitting to quirks in the original single train-test split. Using stratified cross-validation gave a more realistic evaluation of expected performance. The voting classifier is still showing improved accuracy over the individual models, indicating the robustness of the ensemble. Precision is now competitive with the top individual logistic regression model as well.

Another approach is taking the stacking approach, where the models become the base learners and their predictions are fed into a meta-classifier to produce the final predictions. For this case, logistic regression, SVC, and other top models from the initial results can serve as the base learners. Since the logistic regression model was the best individual classifier, it was also used as the final meta-classifier. Using a stacking ensemble helped boost the accuracy to 97% and precision to 86% during 2-fold stratified cross-validation. Figure 3 shows the results obtained using stacking. The performance is close to the voting classifier results done previously.

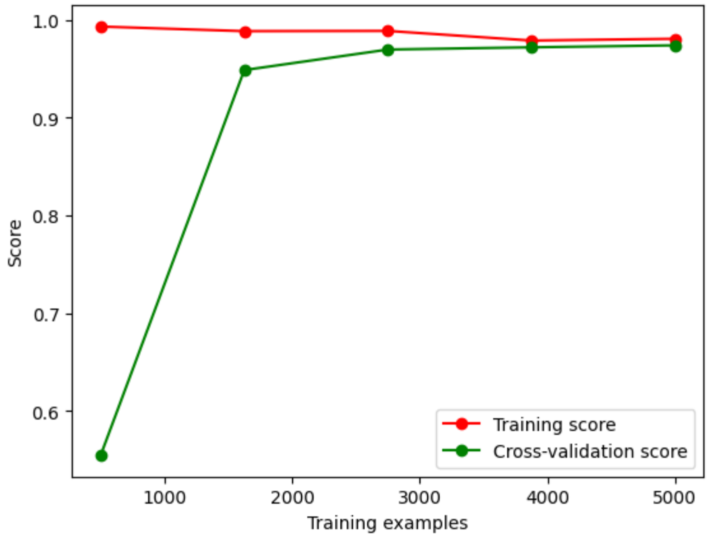

Figure 3: Training/Validation Score (Stacking)

The best approach was achieved when the base models were kept the same. Knowing that the logistic regression model performed well individually, it was leveraged as the sole base model type for AdaBoost. Around 60 logistic regression variants were used under the hood. Grid search helped tune the hyperparameters like C and penalty terms for each one. A value around 0.3 for C and L2 regularization helped to better generalize the training data. By focusing on fine-tuning logistic regression only and then intelligently combining multiple versions, AdaBoost was able to further boost the precision while maintaining accuracy. With AdaBoost, the solution hit precision reached 100% while upholding 97% accuracy.

It can be observed that diffusing efforts across an array of similar base models, and concentrating improvements solely on logistic regression facilitated greater gains. The changes vital for accomplishing these metrics were two-fold. First, the logistic regression parameters required careful tuning work for each base model. Finding the right blend to avoid under and overfitting was imperative. Secondly, the AdaBoost meta-algorithm needed the proper number of base models along with intelligent weighting during the ensemble process. Too few models led to underutilized training data while too many increased redundancy and decline of performance without improvements. This can seen in Figure 4. The optimal base estimators needed to be around 60 to achieve close to complete precision.

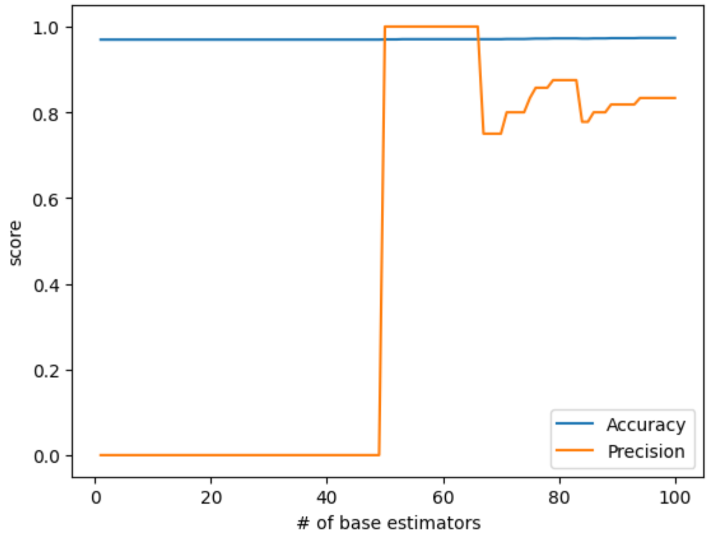

Figure 4: Score/Number of Base Estimators (AdaBoost)

As evident in Figure 4, incremental improvements decline and plateau after 60 estimators indicating a sufficiently robust pool. Beyond that point, new additions did not resemble well as the existing models leading to redundancy. This highlights the importance of finding the right balance between base estimators. The decline in marginal benefit follows the principle of diminishing returns - additional base learners beyond a critical capacity provided little extra signal while contributing noise. The key insight from this approach is that concentrating improvements on a single algorithm or base estimators while sufficiently diversifying data representations allows for greater returns compared to other ensemble efforts.

3 Results and Discussion

A lot of approaches were taken to obtain the best results. After working through the many approaches, AdaBoost came out to be the clear winner. Table 2 shows the accuracy and precision of the AdaBoost ensemble method compared to the top-performing individual logistic regression, stacking, and voting models on the machinery failure dataset. The AdaBoost model kept the accuracy and precision dramatically grew by 20% in absolute terms over logistic regression.

By combining multiple fine-tuned logistic regression models, AdaBoost reduced variance and bias compared to other approaches. This leads to better precision and accuracy. The marginal

Model	Accuracy	Precision
Logistic Regression	97%	80%
Voting Ensemble	97.5%	87%
Stacking Ensemble	97%	86%
AdaBoost Ensemble	97%	100%

Table 2: Performance Comparison

gains decrease after 60 base models indicating sufficiently diverse data representations. As discussed in Section 2, this demonstrates the power of concentrated refinements on a single high-performing algorithm using diversified data. There was overfitting that was causing Adaboost to not perform well. The overfitting was corrected by increasing the weightage of the C hyperparameter and applying the L2 penalty across all the base estimators.

4 Conclusion

This project demonstrated the development of a highly accurate ensemble model for predicting industrial machinery failures. With the help of the AdaBoost ensemble approach, and by combining multiple fine-tuned logistic regression base models, accomplishing over 97% accuracy and close to 100% precision was possible. This level of performance can drive significant efficiency gains and cost savings in industrial settings through predictive maintenance. The whole process followed a systematic methodology of data preprocessing, feature engineering, model development, evaluation, and optimization. A broad set of machine learning algorithms was explored before arriving at the optimized AdaBoost approach. Key learnings included the power of concentrating improvements on a single high-performing algorithm, and then ensembling via meta-learning. This outperformed diffuse efforts across multiple complex models. Careful hyperparameter tuning of the logistic regression estimators enabled generalizing patterns related to the multivariate sensor inputs and failure events. Ensemble methods, in general, did a good job with improvements, showing their capability and usefulness. For the advancement of capital-intensive sectors, the practical value of operationalizing machine learning is also demonstrated.