Predicting Future Machine Failure from Machine State Using Logistic Regression

Matthew Battifarano
Department of Civil and Environmental Engineering
Carnegie Mellon University
mbattifa@andrew.cmu.edu

David DeSmet
Department of Civil and Environmental Engineering
Carnegie Mellon University
ddesmet@andrew.cmu.edu

Achyuth Madabhushi
Department of Civil and Environmental Engineering
Carnegie Mellon University
amadabhu@andrew.cmu.edu

Parth Nabar
Department of Civil and Environmental Engineering
Carnegie Mellon University
pnabar@andrew.cmu.edu

Abstract

Accurately predicting machine failures in advance can decrease maintenance cost and help allocate maintenance resources more efficiently. Logistic regression was applied to predict machine state 24 hours in the future given the current machine state.

1 Introduction

Understanding and predicting machine failure has become a crucial aspect of business operations. 45% of all maintenance efforts are ineffective and 30% of maintenance activities happen too frequently [3]. A data-driven approach to predict machine failure reduces the time and cost burden associated with machine failure. Logistic Regression (LR) is a widely-used data-driven approach to binary classification. LR linearly classifies data into identifiable areas where distance from a decision boundary dictates the probability of inclusion within the class. [8]

Prior literature has also used Support Vector Machines (SVM) to predict failures in server systems [6] and hard-drives [9]. However, SVM, unlike LR, does not provide readily interpretable outcomes that can be provided to a domain expert.

2 Methods

Individual datasets were merged to construct an event stream of machine state. This event stream was used in a logistic regression model to predict the failure of the machine 24 hours in the future. The data was fit using the `LogisticRegression` model provided by the `scikit-learn` library. [7] A

Jupyter notebook[5] with the code used to generate results and figures in this report can be found on GitHub. [2]

2.1 Data processing

The data is composed of several different streams of machine events, identified by the timestamp of the observation, rounded to the hour, and an integer identifier of the machine. These datasets were merged together on observation time and machine identifier to construct an event stream of the complete machine state. Table 1 enumerates the datasets and their schemas.

Table 1: Dataset schemas

dataset	field	description
failures		event stream of component failures
	machine_id	unique machine identifier
	datetime	the observation timestamp
	comp_1	whether or not component 1 failed
	comp_2	whether or not component 2 failed
	comp_3	whether or not component 3 failed
	comp_4	whether or not component 4 failed
errors		event stream of non-failure error codes
	machine_id	unique machine identifier
	datetime	the observation timestamp
	error_1	whether or not error 1 occurred
	error_2	whether or not error 2 occurred
	error_3	whether or not error 3 occurred
	error_4	whether or not error 4 occurred
	error_5	whether or not error 5 occurred
maintenance		component maintenance records
	machine_id	unique machine identifier
	datetime	the maintenance timestamp
	comp_1	whether or not component 1 was replaced
	comp_2	whether or not component 2 was replaced
	comp_3	whether or not component 3 was replaced
	comp_4	whether or not component 4 was replaced
	comp_1_fail	whether or not component 1 was replaced due to failure
	comp_2_fail	whether or not component 2 was replaced due to failure
	comp_3_fail	whether or not component 3 was replaced due to failure
	comp_4_fail	whether or not component 4 was replaced due to failure
telemetry		physical measurements of the machine
	machine_id	unique machine identifier
	datetime	the observation timestamp
	volt	voltage measurements
	rotate	rotation measurements
	pressure	pressure measurements
	vibration	vibration measurements
machines		description of each machine
	machine_id	unique machine identifier
	age	age of the machine in years
	model_1	whether or not the machine is a model 1
	model_2	whether or not the machine is a model 2
	model_3	whether or not the machine is a model 3
	model_4	whether or not the machine is a model 4

The event stream of the complete machine state is constructed by joining the telemetry, maintenance, and errors datasets together on machine_id and datetime. Since telemetry contains measurements every hour for one year, event records are left joined to telemetry. The left join ensures that non-events (e.g. times at which no events occurred) are included in the event stream. The machines dataset is joined in to add age and model type to the data. Finally the day of week is extracted from the timestamp of each data point and included as a categorical feature.

Machine failure is defined as the event that at least one of its four components has failed as reported in the failures dataset. The target variable is defined as the machine failure state 24 hours after the measurement timestamp of each data point in the event steam.

2.2 Logistic regression

Logistic regression is a commonly used classification method known for its easily interpretable model parameters. [1][8] Logistic regression models the probability of a binary target variable Y given the features X by mapping a linear combination of the features to $(0, 1)$ via a non-linear transformation function given by (1).

$$P(Y = 1 \mid X = x) = \pi(x) = \frac{\exp(\alpha + \beta^T x)}{1 + \exp(\alpha + \beta^T x)} \tag{1}$$

In this context, each x is a vector containing the complete machine state of a single machine at a single point in time and the target variable, y, represents the failure state of the machine 24 hours after the measurement. Explicitly, the feature vector is given by Table 2. Note that machine id and timestamp are not included in the feature set, they are simply used as join criteria so that each feature contains data related to the same machine at the same time.

Table 2: The feature vector schema

feature	description
error_1	whether or not error 1 occurred
error_2	whether or not error 2 occurred
error_3	whether or not error 3 occurred
error_4	whether or not error 4 occurred
error_5	whether or not error 5 occurred
comp_1	whether or not component 1 was replaced
comp_2	whether or not component 2 was replaced
comp_3	whether or not component 3 was replaced
comp_4	whether or not component 4 was replaced
comp_1_fail	whether or not component 1 was replaced due to failure
comp_2_fail	whether or not component 2 was replaced due to failure
comp_3_fail	whether or not component 3 was replaced due to failure
comp_4_fail	whether or not component 4 was replaced due to failure
volt	voltage measurements
rotate	rotation measurements
pressure	pressure measurements
vibration	vibration measurements
age	age of the machine in years
model_1	whether or not the machine is a model 1
model_2	whether or not the machine is a model 2
model_3	whether or not the machine is a model 3
model_4	whether or not the machine is a model 4

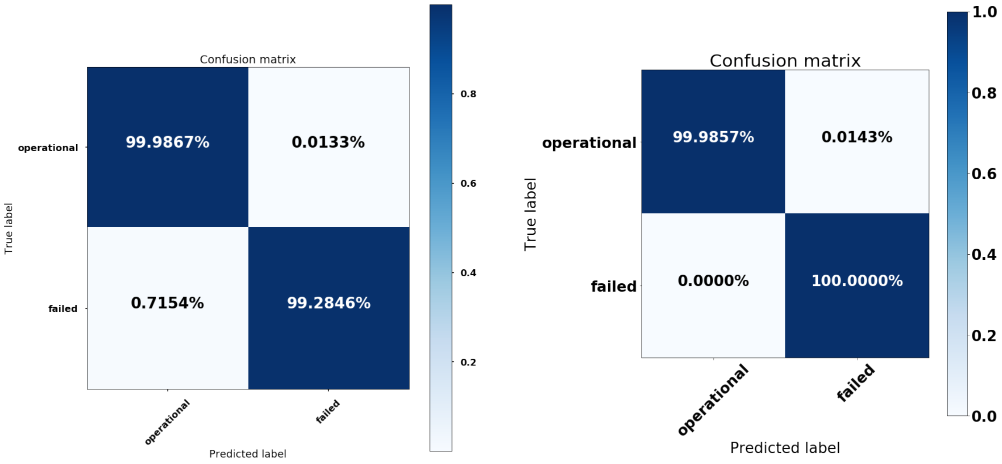
2.3 Model fitting and evaluation

Machine failure is relatively rare: only 1.7% of samples correspond to a failure event. To compensate for the imbalance during training the model should weight the misclassification of a failure event more than misclassification of a non-failure event. To achieve this, each failure sample is given a weight of 100 and each non-failure a weight of 1. A few different values for the weight were tried, but the results appear robust to the sample weight.

To evaluate the model, 3-fold cross validation was employed. To ensure no contamination occurred between the training and testing sets, splits were constructed such that the samples in the testing set occurred chronologically after the samples in the training set and corresponded to different machines. In short, the model was evaluated on future samples from never-before-seen machines.

3 Results

The Logistic regression model performed well on this dataset. On held-out data the trained model was able to perfectly classify all failed events, and nearly perfectly classify all non-failure events.



(a) Normalized average confusion matrix of the validation sets of 3-fold cross validation using the full feature set.

(b) Normalized average confusion matrix of the validation sets of 3-fold cross validation using the reduced feature set.

Figure 1: Normalized average confusion matrices from the model trained on the full feature set (left) and the reduced feature set (right).

The initial formulation of the logistic regression was promising, however, the confusion matrix still had weight on the off-diagonal, predicting machine failures when the machine was still functional or not catching machine failures when they did occur. There was a higher percentage within the confusion matrix for false negatives—that is, when the logistic regression did not accurately predict failures when they occurred, and the team considered this quality something to be avoided.

Additionally, an examination of the weights used in this logistic regression indicated that there was a large proportion of weights that simply were not significant to the model (see 2). These included attribute categories like the day of the week, which component was replaced during maintenance, whether the component replace was due to failure, and the telemetry data. This indicated that these variables could be easily removed from the analysis with little consequence.

It is evident that fewer features affect machine failure than initially understood. The goal is to reduce the frequency of false-negatives by adjusting the size of the β vector to represent fewer, but more important, features. Figure 1b shows the confusion matrix of the logistic regression trained on the smaller feature set. By reducing the feature set to included only the most important features, the accuracy of the logistic regression on the held-out data increased. In particular, it reduced the rate of false negatives where the machine was predicted to be operational but in fact has failed.

4 Discussion

While logistic regression was utilized in this analysis, there are other machine learning techniques that could be applied. For instance, Support Vector Machine classification (SVM) would have a degree of applicability. This is because SVM is easily applied to highly dimensional spaces and our dataset had 30 attributes per observation. [4]

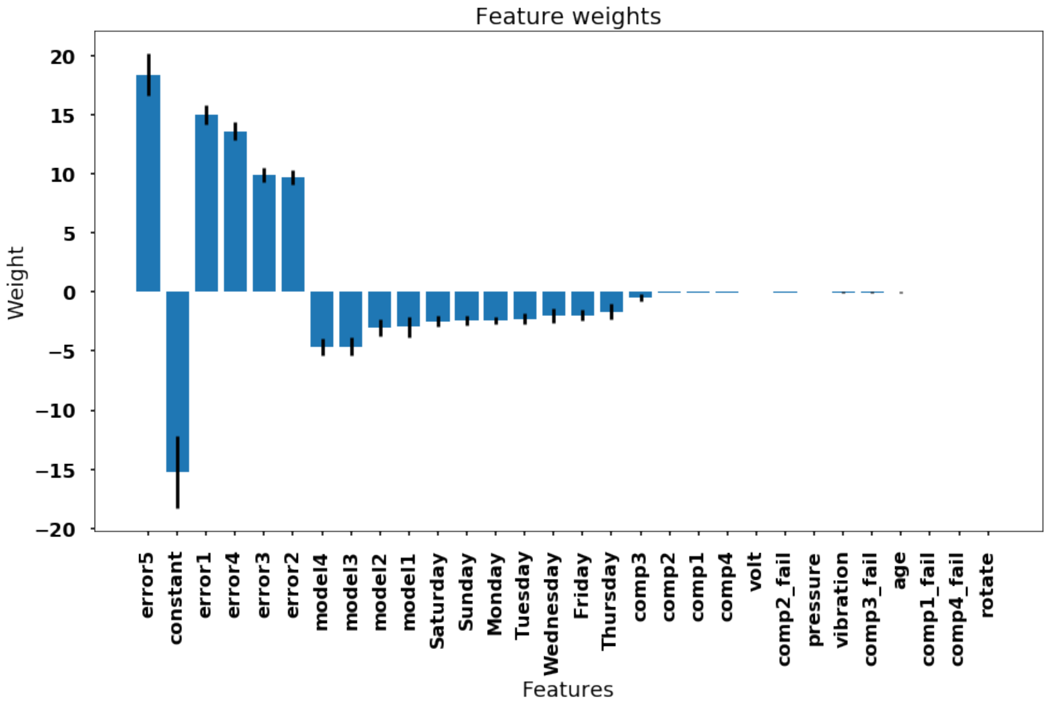


Figure 2: Average values of the feature weights over 3-fold cross validation ordered by magnitude. The 'constant' feature corresponds to α in (1) while the rest correspond to elements of β.

In terms of improvement on the logistic regression, the largest potential is with false positives. That is, while demonstrating (within the test set) 100% accuracy when it came to predicting failures, the logistic regression predicted failures in machines that were still operational. Thankfully, this is most likely where the margin of safety should be. That is, a machine failing before being caught is most likely more detrimental than a machine being inspected in a controlled environment. Nevertheless, this still is the area where our model is most in need of improvement.

One issue to explore is how sensitive logistic regression is to the time alloted before the point of failure. Currently, a 24 hour period is applied, it is interesting to attempt the logistic regression for both a shorter and longer periods. Therefore, further analysis could investigate whether or not larger periods of time would have comparable (or perhaps even improved) accuracy. Additionally, this additional analysis could incorporate a breadth of data over time.

The current logistic regression does not distinguish between components when a failure is predicted, just machine failure. Thus, further analysis might look into predicting which components will likely fail. Pinpointing component failure may further decrease maintenance costs, as the maintenance team would no longer need to have repairing capabilities for all of the components, just specific failure-prone components.

5 Acknowledgments

Our deepest gratitude goes to the Auton Lab at Carnegie Mellon University for providing the data set, and organizing the HackAuton. Additional thanks to the sponsors for providing the funding and datasets to make this HackAuton possible.

References

[1] Alan Agresti. Logistic regression. In *Categorical Data Analysis*, chapter 5, pages 165–210. Wiley-Blackwell, 2003.

[2] DeSmet D. Madabhushi A. Nabar P. Battifarano, M. Predicting future machine failure from machine state using logistic regression. `https://github.com/mbattifarano/2018.hackAuton`, 2018.

[3] IBM corp. Are your preventive maintenance efforts wrenching away precious time and resources?

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[5] Thomas Kluyver, Benjamin Ragan-kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter Notebooks—a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90, 2016.

[6] Joseph F Murray, Gordon F Hughes, and Kenneth Kreutz-Delgado. Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. *J. Mach. Learn. Res.*, 6:783–816, 2005.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Cosma Rohilla Shalizi. Logistic Regression. In *Advanced Data Analysis from an Elementary Point of View*, chapter 12 Logistic Regression. Cambridge University Press, 2012.

[9] Doug Turnbull and Neil Alldrin. Failure prediction in hardware systems. Technical report, 2003.