Predictive Maintenance with a Milling Machine Use Case

11.2

February 2026

DOCUMENT ACCESS

Public

DISCLAIMER

The contents of this document are under copyright of Critical Manufacturing S.A. it is released on condition that it shall not be copied in whole, in part or otherwise reproduced (whether by photographic, or any other method) and the contents therefore shall not be divulged to any person other than that of the addressee (save to other authorized offices of his organization having need to know such contents, for the purpose for which disclosure is made) without prior written consent of submitting company.

Predictive Maintenance with a Milling Machine Use Case

Estimated time to read: 5 minutes

The information that follows serves to guide you on how to implement a predictive maintenance workflow using machine learning.

Overview

Predictive maintenance uses machine data to predict when a failure is likely to occur, which enables proactive maintenance to minimize downtime and reduce costs. In this tutorial, we will demonstrate how to implement predictive maintenance for a milling machine, leveraging insights gathered during data collection.

This use case focuses on the following key failure scenarios:

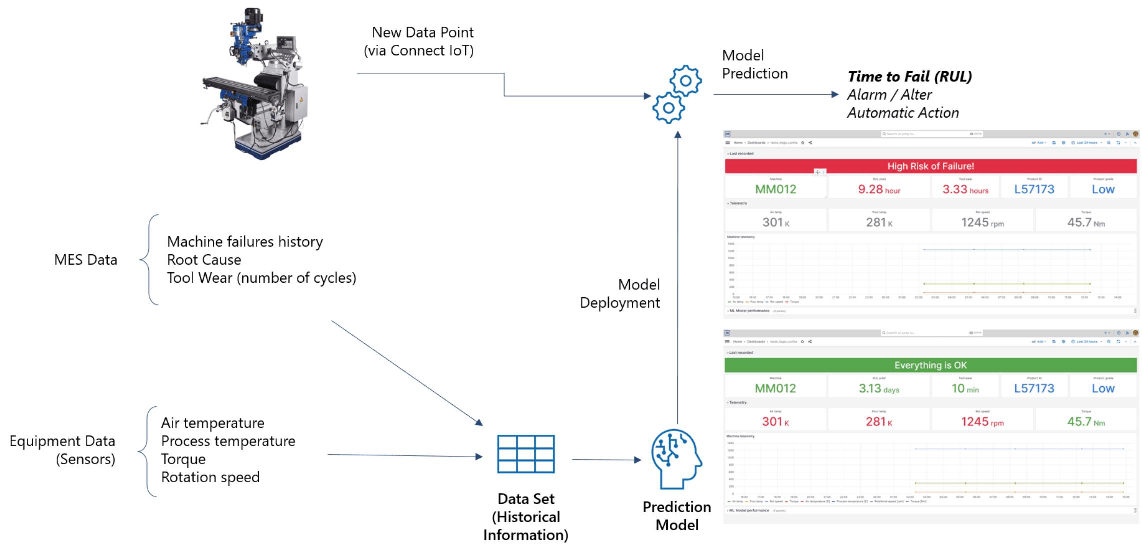
1. **Tool Wear Failure (TWF)**
2. **Heat Dissipation Failure (HDF)**
3. **Power Failure (PWF)**
4. **Overstrain Failure (OSF)**
5. **Random Failures (RNF)**

We will also estimate the Remaining Useful Life (RUL) of the milling machine based on observed time intervals between failures.

Preconditions

Before implementing predictive maintenance, ensure the following:

1. **Data Collection**:
    - Historical failure data with features such as tool wear, temperature, rotational speed, torque, and power usage.
2. **Analytical Tools**:
    - A data visualization platform (example: Grafana) and access to Data Platform ML Model.



Steps to Implement Predictive Maintenance

As context, let's consider a milling machine with the following failure scenarios:

- **TWF**: Tool wear reaches its limit (200-240 min).
- **HDF**: Heat dissipation fails when temp difference < 8.6 K and rotational speed < 1380 rpm.
- **PWF**: Power fails when P < 3500 W or P > 9000 W.
- **OSF**: ToolWear x Torque > 11000 minNm for load L, 12000 for M, 13000 for H.
- **RNF**: Rare random failures (<1%).

Step 1: Data Preprocessing

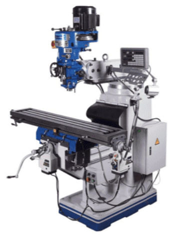
1. **Import Data**: Load the data via Post Event and store it in a Data Set. For more information, see Outlier Detection using Machine Learning. Ensure it contains fields you need.
2. **Clean Data**: Handle missing values and outliers. Standardize or normalize data for consistent analysis. This can be done for you in the ML Model wizard.
3. **Feature Engineering**:
   - Optionally, we may compute derived metrics like power using P = Torque * Rot. Speed (rad/s).

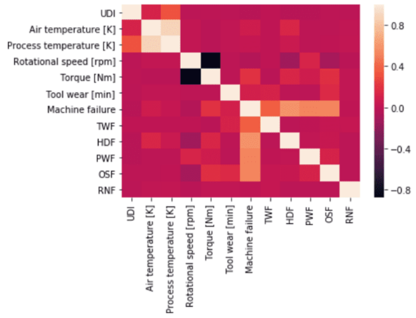
> **Info**
>
> Once again, these last steps may be performed within the ML Model.

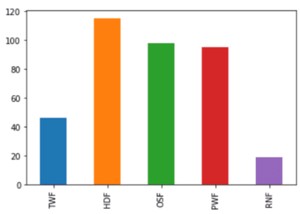
Step 2: Exploratory Data Analysis

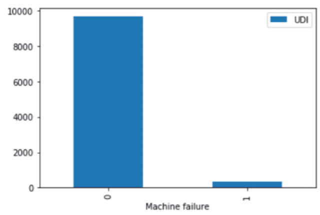
1. **Visualize Data**: Use graphs and dashboards to identify trends and patterns.
   - Example: Plot tool wear over time to observe failure intervals.
2. **Correlations**: Analyze correlations between failure events and parameters like tool wear, temperature, and power.
3. **Failure Insights**: Validate that failure conditions align with the thresholds provided:
   - Example: Check if failures occur when temp difference < 8.6 K and rotational speed < 1380 rpm (HDF).

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	TWF	HDF	PWF	OSF	RNF
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	0	0	0	0	0







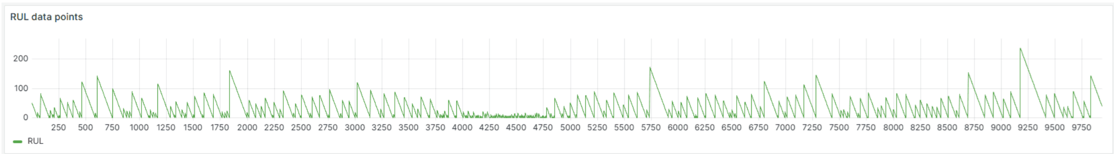


In the image, we can see the data structure along with their correlation matrix. For this particular dataset, we observe the dataset is not balanced - for classification problems, you may first consider apply over-sampling and/or under-sampling techniques.

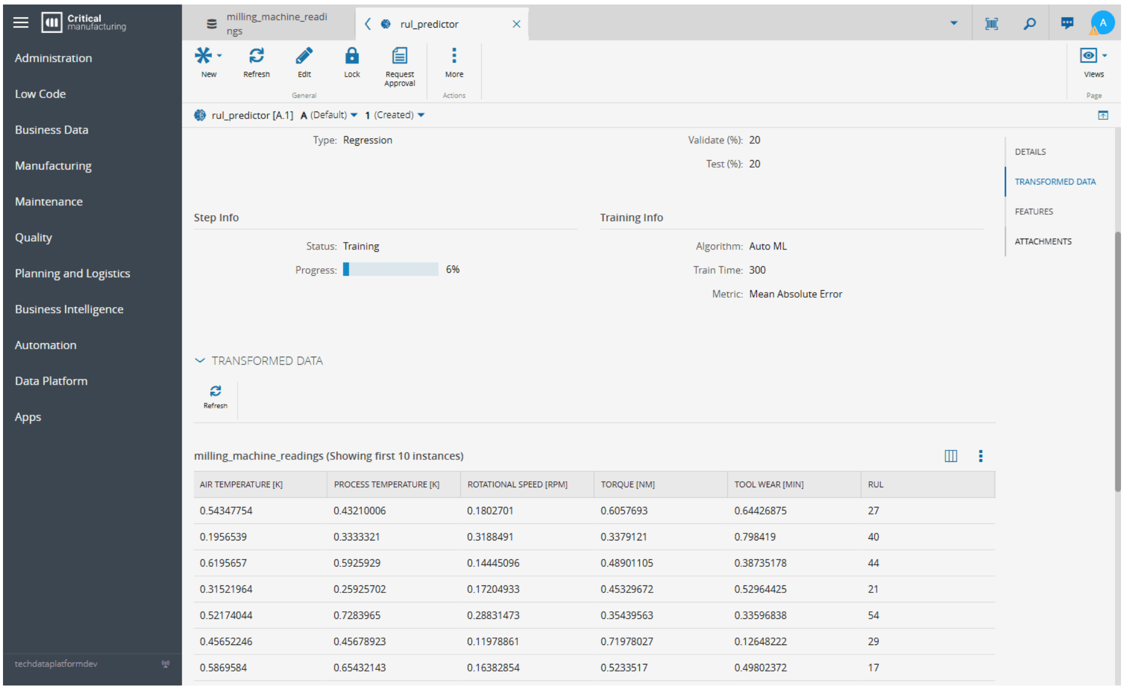
Step 3: Predictive Modeling

Let's divide the predictive modeling into two parts:

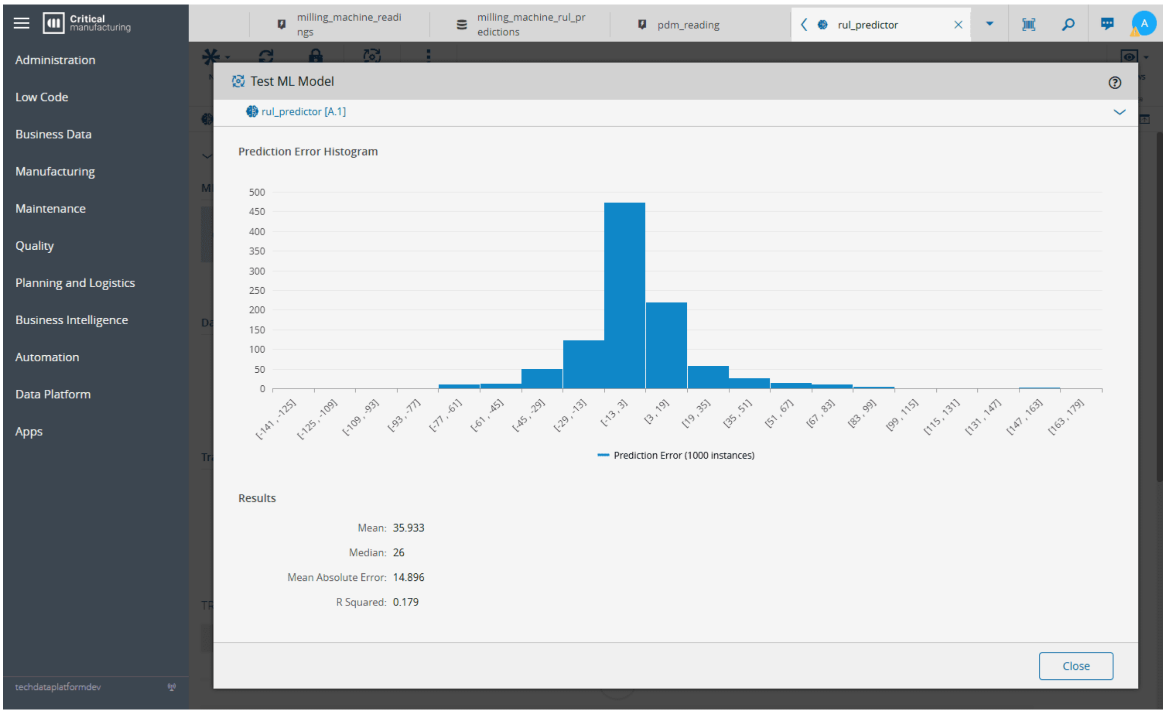
1. **Label RUL**:
   - Calculate the time remaining until the next failure for each data point.
   - Use a linear approximation based on the observed time intervals between failures.



2. **Train RUL Model**:
   - Train the model using features like tool wear, temperature, power, and torque. Select, during the ML Model wizard, the RUL column as label. This will define the experiment as a regression problem - you can check the Details view later on.
   - Use the ML Model to predict RUL. Validate the ML Model's performance.



As this is a regression problem, choose a metric that fits your needs. When it is trained, you only need to set it effective to deploy the model in MES ecosystem.

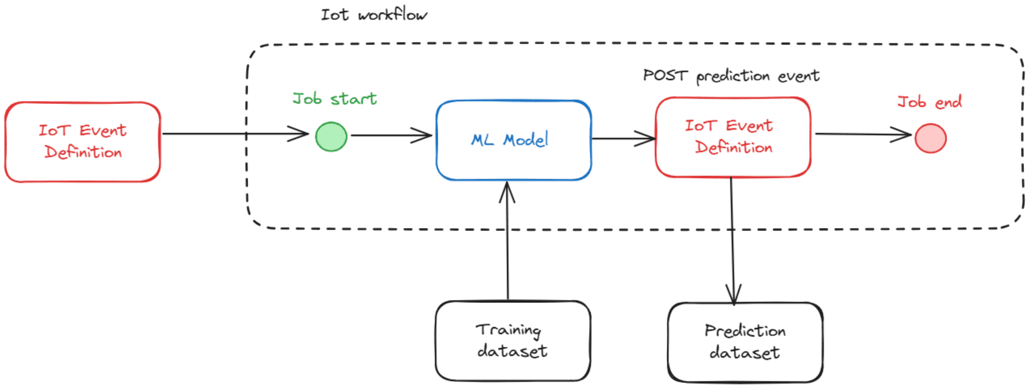


> **Info**
>
> Remember that the ML Model consist in an AutoML framework, thus, it will automatically select the best model for your problem. For more information, see this link.

Step 4: Integrate with Real-Time Monitoring

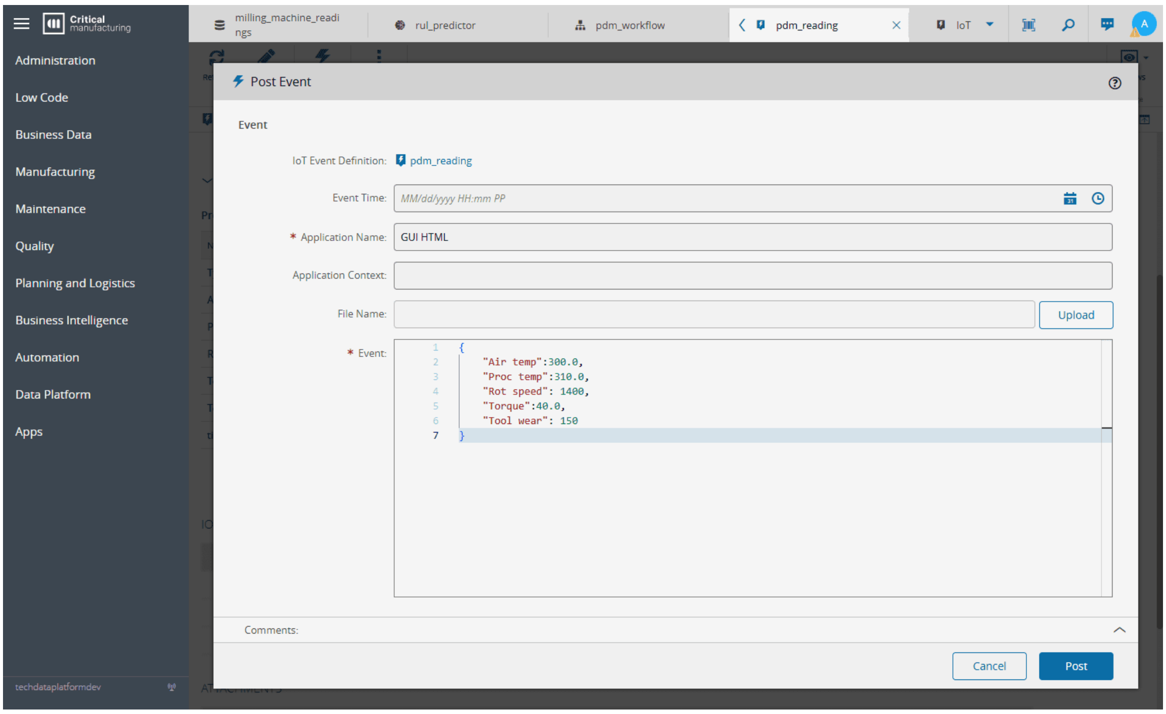
1. **Set Up IoT workflow**:

- Assign an IoT workflow to post the readings. The workflows, when triggered, will then predict the time to fail and record the value in a Data Set via Post Event, as shown in the image:



2. **Add ML Task**:
   - Drag the ML Task and connect the inputs, but first you must select the right IoT Event Definition - select the settings button to do so. For more information, see Outlier Detection using Machine Learning.
3. **Add API Post Event Task**:
   - Add API Post Event Task to record the predicted RUL in a Data Set.
   - Select the settings button to choose the right event definition.

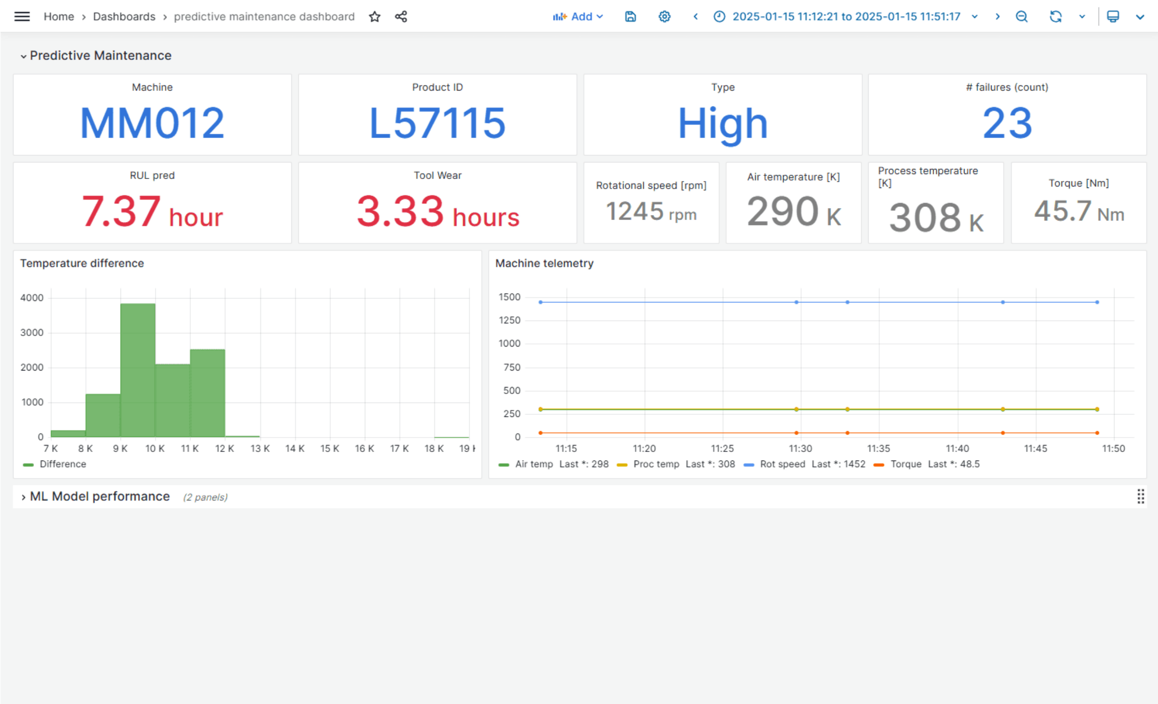
When this is done, connect the blocks and run the workflow by manually posting an event. Example:



You should see a new record in the predicted RUL Data Set.

Step 5: Validate and Optimize

1. **Create Dashboards**:
   - Display real-time RUL estimates, failure type predictions, and key parameters (example: tool wear, power).
2. **Test Predictions**:
   - Compare model predictions with actual failures to evaluate accuracy.
3. **Adjust Models**:
   - Refine thresholds and retrain models with updated data to improve predictions.
4. **Monitor Performance**:
   - Continuously monitor the workflow to ensure reliability.



Final remarks

The example dashboard for the milling machine could include:

- **Tool Wear Monitor**: Visualize tool wear over time, highlighting when it nears failure thresholds.
- **Telemetry Trends**: Show temperature differences and power consumption to detect anomalies.
- **RUL Estimation**: Display predicted RUL in minutes.

By following this tutorial, you can build a predictive maintenance workflow for a milling machine that improves uptime, reduces costs, and ensures smooth production processes. For further assistance, refer to your data analytics platform documentation or consult with your engineering team.

Legal Information

Disclaimer

The information contained in this document represents the current view of Critical Manufacturing on the issues discussed as of the date of publication. Because Critical Manufacturing must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Critical Manufacturing, and Critical Manufacturing cannot guarantee the accuracy of any information presented after the date of publication. This document is for informational purposes only.

Critical Manufacturing makes no warranties, express, implied or statutory, as to the information herein contained.

Confidentiality Notice

All materials and information included herein are being provided by Critical Manufacturing to its Customer solely for Customer internal use for its business purposes. Critical Manufacturing retains all rights, titles, interests in and copyrights to the materials and information herein. The materials and information contained herein constitute confidential information of Critical Manufacturing and the Customer must not disclose or transfer by any means any of these materials or information, whether total or partial, to any third party without the prior explicit consent by Critical Manufacturing.

Copyright Information

All title and copyrights in and to the Software (including but not limited to any source code, binaries, designs, specifications, models, documents, layouts, images, photographs, animations, video, audio, music, text incorporated into the Software), the accompanying printed materials, and any copies of the Software, and any trademarks or service marks of Critical Manufacturing are owned by Critical Manufacturing unless explicitly stated otherwise. All title and intellectual property rights in and to the content that may be accessed through use of the Software is the property of the respective content owner and is protected by applicable copyright or other intellectual property laws and treaties.

Trademark Information

Critical Manufacturing is a registered trademark of Critical Manufacturing.

All other trademarks are property of their respective owners.

Copyright © 2023 Critical Manufacturing. All rights reserved.