

PROJECT TYPE

AI-powered predictive maintenance solution

TECHNOLOGIES

IoT: OPC UA, MQTT;
Data: TimescaleDB, AWS S3;
ML: Python, scikit-learn/XGBoost;
Dashboards: React, Grafana.

DURATION

6 months

METHODOLOGY

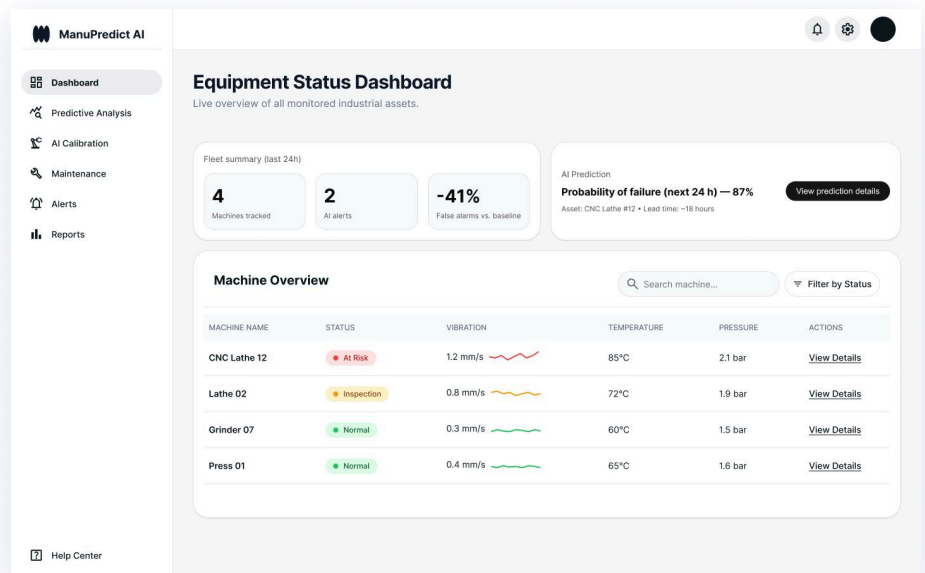
Scrum

TEAM

1 Product Manager
 1 Business Analyst
 2 Data Scientists
 1 Backend Engineer
 1 Frontend Engineer
 1 DevOps Engineer

AI-powered predictive maintenance platform for a large industrial manufacturer

An upgrade of an existing IoT system with explainable machine learning models and context-aware analytics. The solution predicts equipment failures earlier, minimizes false alarms, and integrates directly with maintenance workflows to improve production uptime.



The Client is a large industrial manufacturer in Central Europe producing machinery components across multiple plants. A few years ago, we delivered an IoT foundation that streamed vibration, temperature, and pressure data from shop-floor equipment into dashboards and alerts. The system reduced unplanned downtime and enabled planned maintenance.

Project Distinctive Features

- ✓ Hybrid prediction engine – combined supervised and anomaly-detection models for better accuracy on both known and new equipment types.
- ✓ Adaptive calibration – weekly retraining with automatic drift detection to sustain model quality.
- ✓ Context-aware logic – incorporated workload, ambient temperature, and operational schedules.
- ✓ Explainable interface – displayed clear contributing factors behind each alert.

- ✓ CMMS integration – created prioritized maintenance tasks directly from AI predictions.
- ✓ Edge fallback – ensured continuous operation during network disruptions.

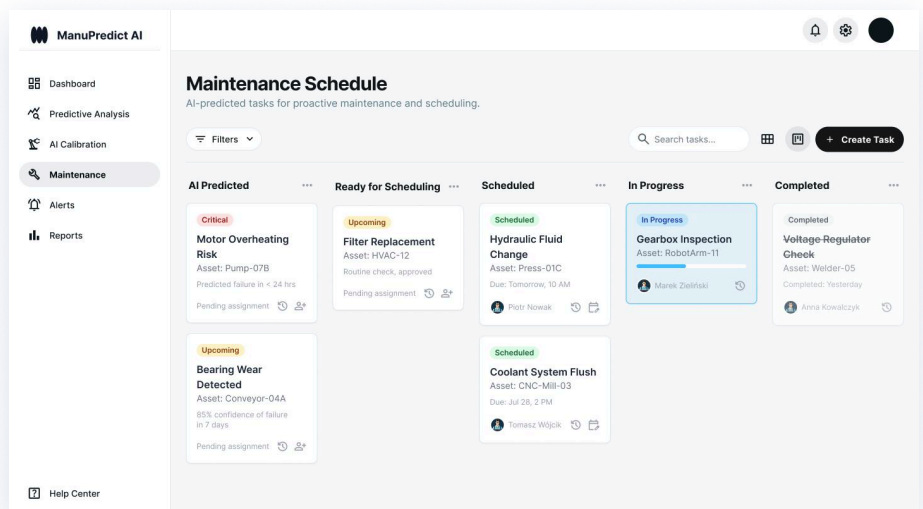
Business challenge

As production scaled and new machine models were introduced, prediction accuracy fell and alert noise rose. The existing system couldn't capture complex parameter interactions (e.g., ambient temperature × workload).

The Client wanted to move from basic failure warnings to an adaptive, context-aware model that learns from new data and predicts issues earlier.

Our solution

We augmented the existing IoT platform with an explainable ML layer, tied predictions to the Client's maintenance workflow, and set up light MLOps for continuous calibration. Work happened without stopping production: we ran an 8-week pilot on selected assets, then rolled out in 6 phases, line by line.



How our AI model works

Data foundation – multi-year sensor history (~20 TB) served as the training base. The data pipeline aggregates vibration, temperature, and pressure signals into time windows and computes features such as RMS, kurtosis, and spectral energy peaks used for model training.

Model architecture – we combined two complementary approaches:

- ✓ Gradient-boosted trees (XGBoost) trained on labeled failure cases to predict risk scores for known equipment types.
- ✓ Anomaly detection module (isolation forest + autoencoder) to flag unusual behavior in machines with few or no historical failure labels.

These models operate together: the anomaly module detects deviations early, and the supervised model confirms or rejects them, reducing false positives.

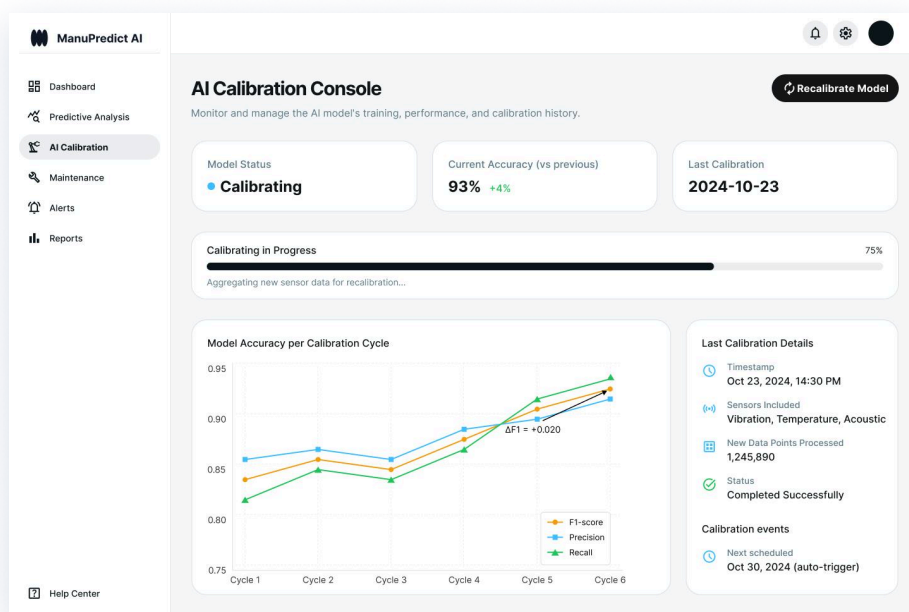
Context enrichment – along with physical sensor data, the model ingests ambient temperature, workload intensity, and operating shifts. This allows predictions to reflect real production conditions rather than static thresholds.

Continuous learning – weekly recalibration updates model weights using the latest data. Drift detection monitors feature distributions, and if deviation exceeds a set threshold, retraining is triggered automatically.

Explainability – each prediction exposes top contributing factors (for example, “vibration ↑ 18% over baseline” or “temperature × load correlation ↑ 0.35”). Engineers can trace every alert to specific sensor patterns, improving trust and adoption.

Integration – prediction results are passed to the Client’s CMMS via API. Each alert generates a prioritized maintenance task with asset ID, failure probability, and lead-time estimate.

Deployment design – inference runs in the cloud, while edge gateways cache the latest model for offline work. Updates are delivered through blue-green deployments, ensuring zero downtime.



Development phases

Another goal we had was to keep production stable while developing and testing the AI module. After confirming the model’s accuracy and lead-time prediction during the pilot, we expanded the core solution through six consecutive phases.

Phase 1: assess & align – clarified failure modes, maintenance playbooks, data quality, and success criteria with the Client.

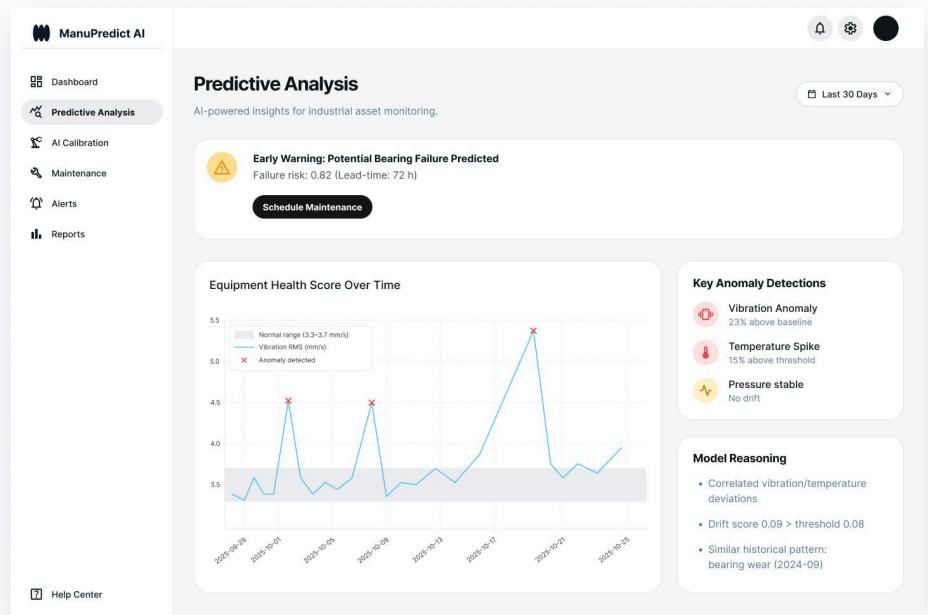
Phase 2: model the risk – built a feature pipeline from vibration/temperature windows and trained supervised + anomaly models.

Phase 3: calibrate & monitor – tuned thresholds to the cost of errors, set weekly recalibration, and added drift monitoring.

Phase 4: integrate with work orders – connected predictions to the Client’s CMMS to auto-create prioritized jobs with lead-time windows.

Phase 5: dashboards & UX – delivered equipment health, predictive analysis, calibration console, and a maintenance board for planners.

Phase 6: safe rollout – used blue-green deployments, edge fallback for unstable links, and access controls aligned with the Client's policies.



Customer's benefits

False alerts decreased by over 40%, while early failure detection increased by 35%. Unplanned downtime was reduced by approximately 50%, saving around \$274,000 in eight months and delivering a 2.5x ROI.

What's happening with the project right now?

The solution now runs across 20 production lines and continues to recalibrate automatically as new data arrives.