

CAPSTONE PROJECT

Predictive Maintenance of Industrial Machinery

(Problem statement No. 39)

Presented By:

KARUNAMAYUDU DOPPALAOUDI

**RISE KRISHNA SAI Prakasam Group of
Institutions**

Computer Science and Engineering

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach
(Technology Used)**
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

In modern industrial sectors, unplanned equipment failures can lead to costly downtime and decreased productivity. To address this, many facilities have started integrating real-time sensor technologies in machines to monitor their health. However, identifying the exact moment and type of potential machine failure remains a major challenge. Predicting failures in advance allows for timely maintenance scheduling and uninterrupted operations. The critical task is to develop a model that can predict the failure type (e.g., tool wear, power failure, heat dissipation issues) using real-time sensor data from the machinery. This ensures proactive maintenance, reduces unexpected downtimes, and optimizes operational efficiency.

PROPOSED SOLUTION

The proposed system aims to address the challenge of predicting the type of machinery failure in advance to enable timely and proactive maintenance actions. This involves leveraging sensor data and machine learning techniques to classify failure types accurately. The solution will consist of the following components:

Data Collection:

- Gather historical and real-time sensor data from various industrial machines, including attributes such as rotational speed, torque, tool wear, and vibration levels.
- Use the publicly available dataset from Kaggle: Machine Predictive Maintenance Classification, which contains labeled failure types like heat dissipation, power failure, and tool wear.

Data Preprocessing:

- Clean and preprocess the dataset to address missing values, outliers, and data imbalance issues.
- Perform normalization or standardization of sensor readings to ensure consistency.
- Engineer relevant features from the time-series data that reflect underlying machine behavior and degradation patterns.

Machine Learning Algorithm:

- Implement a classification algorithm (e.g., Snap Random Forest Classifier, XGBoost, or Support Vector Machine) to predict the type of failure based on sensor input.
- Optionally explore deep learning models such as LSTM or GRU if temporal dependencies in the data are significant.
- Perform hyperparameter tuning and cross-validation to ensure robust model performance.

Deployment:

- The selected model was deployed using IBM Watson Studio's AutoML deployment feature.
- The system supports: Input via manual CSV or JSON, Instant predictions through the deployed API interface.

Evaluation:

- Evaluate the model using metrics such as Accuracy, Precision, Recall, F1-score, and Confusion Matrix to measure classification performance.
- Perform real-world simulation or testing in a controlled environment to assess model responsiveness and reliability.
- Continuously monitor the model in production and retrain it periodically with new data to maintain high accuracy.

Result:

- The proposed system enables early detection of machinery faults, helping industries schedule maintenance activities proactively. This reduces unexpected breakdowns, minimizes production losses, lowers maintenance costs, and extends machine life. By providing actionable insights in real-time, the system supports data-driven maintenance decisions for industrial efficiency.

SYSTEM APPROACH

This section outlines the methodology and tools employed to develop and deploy the predictive maintenance model, aimed at forecasting equipment failures in industrial machinery using real-time sensor data.

System Requirements

| Component | Specification |
|---------------------|---|
| Platform | IBM Watsonx.ai Studio (Cloud-based) |
| Dataset Source | Kaggle: Predictive Maintenance Classification |
| Processor | Cloud-based, managed by IBM infrastructure |
| Storage | IBM Object Storage / Watson Studio Asset Storage |
| Interface | Web-based User Interface (via Watson Studio Deployment) |
| File Formats | CSV for input; JSON supported for prediction API |
| Max Input File Size | 50 MB |

Libraries and Tools Used

| Tool / Library | Purpose |
|---------------------------|--|
| IBM Watson Studio | Model development, training, AutoAI pipelines |
| AutoAI | Automated model building, evaluation, and optimization |
| Scikit-learn (backend) | Classification algorithms (e.g., Random Forest) |
| Pandas & NumPy | Data preprocessing, feature extraction |
| Matplotlib / Seaborn | Visualization of training/evaluation metrics |
| Watson Machine Learning | Deployment of final model as a web service |
| REST API (auto-generated) | Used for integrating with front-end applications |

ALGORITHM & DEPLOYMENT

This section outlines the machine learning algorithm used to predict the type of machinery failure based on real-time sensor inputs. The algorithm was selected and optimized using IBM Watson AutoAI, which automatically compares multiple models and pipelines to identify the best-performing solution.

Algorithm Selection

The chosen algorithm for this project is the **Batched Tree Ensemble Classifier**, implemented as a Snap Random Forest Classifier within IBM Watson AutoAI. This algorithm was selected after evaluating multiple pipelines, including Decision Tree Classifiers and standalone Random Forest models.

Random Forest is well-suited for this classification problem because it:

- Handles high-dimensional, mixed-type input data (numerical + categorical)
 - Is robust to outliers and noise
 - Offers high interpretability and generalization performance
- It achieved an optimized accuracy of 99.5% during cross-validation, outperforming other contenders in precision, recall, and F1-score.

Data Input

| Feature Name | Description |
|-------------------------|-------------------------------------|
| UDI | Unique identifier for each data row |
| Product ID | Unique ID of the machine model |
| Type | Machine type (H, M, L) |
| Air Temperature [K] | Environmental temperature |
| Process Temperature [K] | Machine internal temperature |
| Rotational Speed [rpm] | Motor RPM |
| Torque [Nm] | Rotational force applied |
| Tool Wear [min] | Tool usage duration |

Training Process

- The dataset was split into training and holdout sets using **AutoAI's automatic data partitioning**.
- Preprocessing steps included normalization, encoding categorical variables, and feature selection.
- **Hyperparameter optimization (HPO)** was performed to fine-tune the Random Forest model parameters, such as:
 - Number of estimators
 - Max tree depth
 - Feature sampling method
- **Cross-validation** was applied to ensure that the model generalizes well to unseen data.
- Multiple pipelines (P1–P9) were evaluated, and the best-performing one (P5) was selected based on overall accuracy and log loss.

Prediction Process

Once deployed, the model accepts **real-time or batch input** via CSV or JSON format through a web interface or API. The model processes the input features and predicts the **failure type**, which may include:

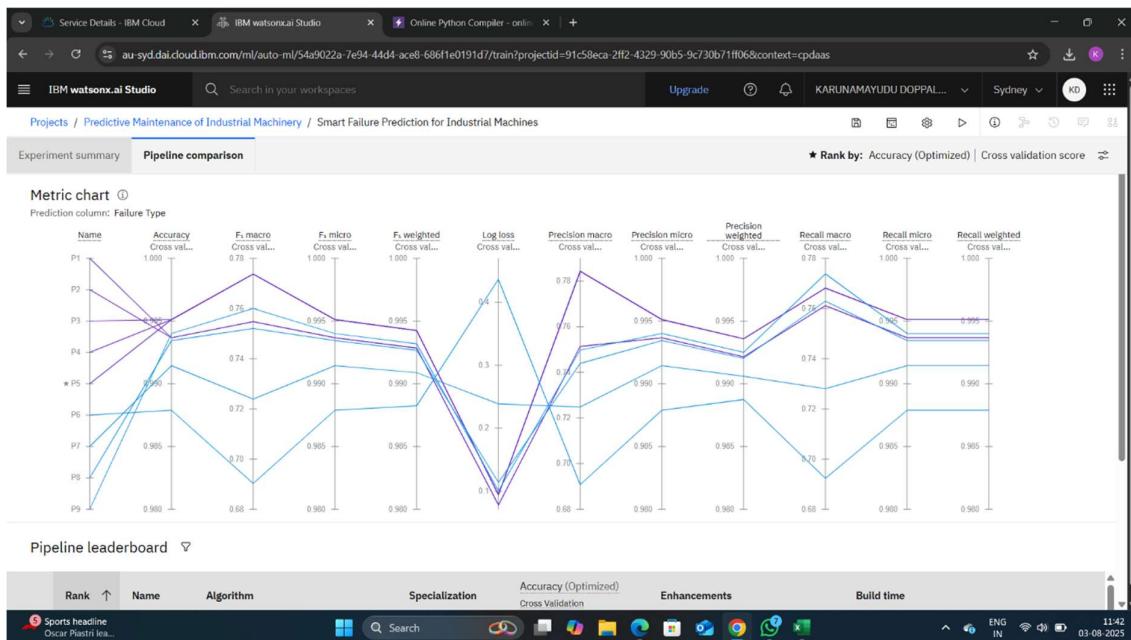
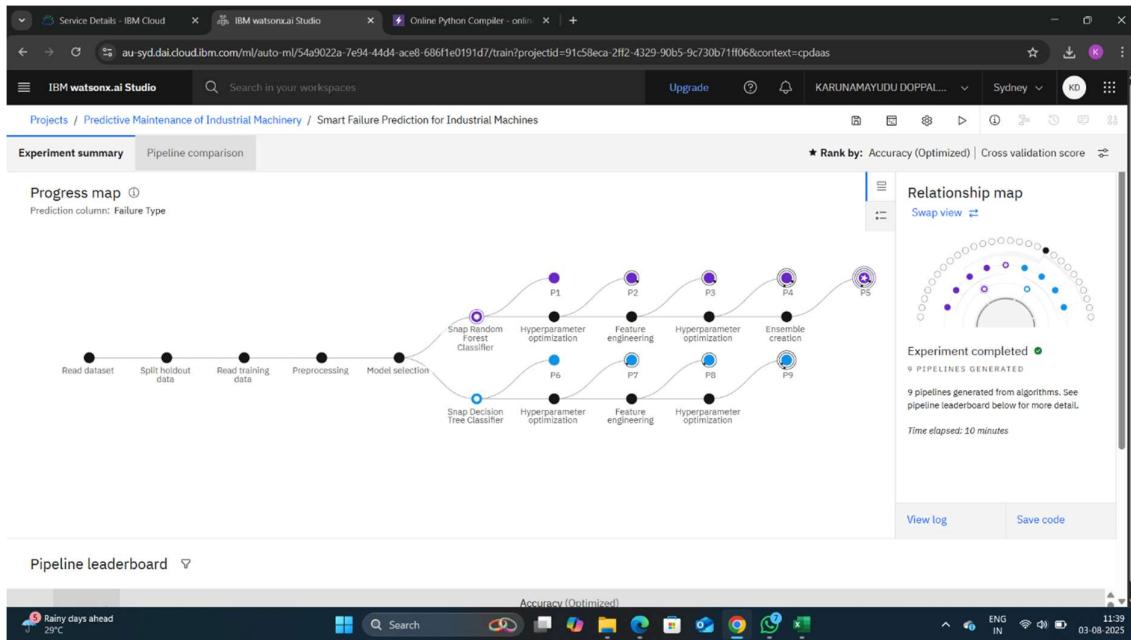
- Heat Dissipation Failure
- Tool Wear Failure
- Power Failure
- Overstrain Failure
- No Failure (Normal operation)

The real-time system can be integrated into industrial monitoring dashboards, enabling instant alerts for preventive maintenance actions based on the predicted failure category.

RESULT

The screenshot shows the 'Configure AutoAI experiment' screen in IBM Watson AI Studio. On the left, there is a 'Add data source' panel with a 'predictive_mainaintenance.csv' file selected. On the right, the 'Configure details' panel is open, showing settings for predicting 'Failure Type'. The prediction column is set to 'Failure Type', and the prediction type is 'Multiclass Classification'. The experiment is optimized for 'Accuracy & run time'. A progress bar indicates 'CUH remaining: 20 CUH'. At the bottom, there are 'Experiment settings' and a 'Run experiment' button.

The screenshot shows the 'Experiment summary' screen in IBM Watson AI Studio. It features a 'Relationship map' visualization with three concentric rings labeled 'FEATURE TRANSFORMERS', 'PIPELINES', and 'TOP ALGORITHMS'. Below the map is a 'Pipeline leaderboard' table. To the right, a 'Progress map' shows the status of pipeline generation, stating 'Experiment completed' and '9 PIPELINES GENERATED'. The time elapsed is 10 minutes. At the bottom, there are 'View log' and 'Save code' buttons.



Pipeline summary

Pipeline comparison

Experiment summary

Rank by: Accuracy (Optimized) | Cross validation score

Pipeline leaderboard

| Rank | Name | Algorithm | Specialization | Accuracy (Optimized) | Enhancements | Build time |
|------|------------|---|----------------|----------------------|----------------------|------------|
| ★ 1 | Pipeline 5 | Batched Tree Ensemble Classifier (Snap Random Forest Classifier) | INCR | 0.995 | HPO-1 FE HPO-2 BATCH | 00:02:25 |
| 2 | Pipeline 4 | Snap Random Forest Classifier | | 0.995 | HPO-1 FE HPO-2 | 00:01:32 |
| 3 | Pipeline 3 | Snap Random Forest Classifier | | 0.995 | HPO-1 FE | 00:01:22 |
| 4 | Pipeline 9 | Snap Decision Tree Classifier | | 0.994 | HPO-1 FE HPO-2 | 00:00:03 |

Predictive Maintenance of Industrial Machinery Model Deployed Online

Test

Enter input data

Text **JSON**

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

Download CSV template ↴ Browse local files ⌂ Search in space ↵ Clear all ✕

| | UDI (double) | Product ID (other) | Type (other) | Air temperature [K] (double) | Process temperature [K] (double) | Rotational speed [rpm] (double) | Torque [Nm] (double) | Tool wear |
|---|--------------|--------------------|--------------|------------------------------|----------------------------------|---------------------------------|----------------------|-----------|
| 1 | 14 | M14873 | M | 298.6 | 309.2 | 1742 | 30 | 37 |
| 2 | 51 | L47230 | L | 298.9 | 309.1 | 2861 | 4.6 | 143 |
| 3 | 101 | H29400 | H | 305.6 | 350.6 | 1500 | 99 | 145 |
| 4 | 1001 | M4168 | H | 250.67 | 450.563 | 3000 | 205 | 400 |

4 rows, 9 columns

Predict

Service Details - IBM Cloud Predictive Maintenance of Indi... Online Python Compiler - onlin...

IBM Watson AI Studio Search in your workspaces

Deployment spaces / Predictive Maintenance of Industrial Machinery Using ML / P5 - Snap Random Forest Classifier: Smart Failure Prediction for Industrial Machines /

Prediction results

Prediction type: Multiclass classification

Prediction percentage:



4 records

No Failure Power Failure

Display format for prediction results
 Table view JSON view

Show input data

| | Prediction | Confidence |
|----|---------------|------------|
| 1 | No Failure | 100% |
| 2 | Power Failure | 100% |
| 3 | Power Failure | 90% |
| 4 | Power Failure | 60% |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |

Download JSON file

29°C Haze Search Home File Edit View Insert Tools Help

ENG IN 11:57 03-08-2025

CONCLUSION

This project successfully developed and deployed a machine learning-based predictive maintenance system for industrial machinery using real-time sensor data. Leveraging IBM Watson's AutoAI capabilities, the most optimal model—a Batched Tree Ensemble Classifier—was identified, achieving a high classification accuracy of **99.5%**. The model effectively predicts potential failure types such as tool wear, heat dissipation, and power failure, enabling timely and proactive maintenance decisions.

Throughout the implementation, challenges included managing data preprocessing, handling imbalanced failure classes, and optimizing hyperparameters for best performance. These were effectively addressed using AutoAI's automated pipeline generation and cross-validation techniques.

While the current system performs exceptionally well, future improvements could include integrating real-time streaming data, adaptive learning for continuous updates, and enhanced visualization dashboards for industrial teams.

Accurate failure prediction plays a crucial role in reducing machinery downtime, optimizing maintenance schedules, and cutting operational costs—similar in impact to how accurate bike count predictions support urban mobility planning. Just as stability in bike supply ensures rider satisfaction and citywide efficiency, early fault detection in machines ensures smoother industrial operations and higher productivity.

FUTURE SCOPE

In the future, the predictive maintenance system can be enhanced by incorporating additional sensor data such as vibration analysis, humidity, voltage fluctuations, and machine-specific operational logs. Integrating real-time data streams from IoT-enabled industrial devices can improve prediction accuracy and help detect failures even earlier. The model can also be continuously retrained using live data to adapt to changing machine behaviors, ensuring its relevance and accuracy over time.

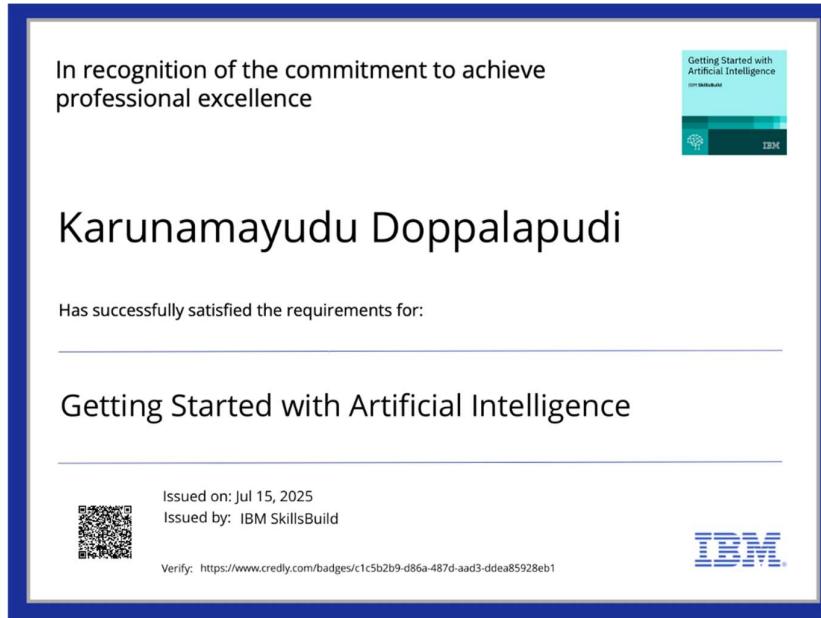
Further improvements may include the use of advanced machine learning techniques such as deep neural networks (e.g., LSTM, CNN) or ensemble hybrid models to capture complex patterns in time-series sensor data. The system can be extended to support edge computing, allowing predictions to happen directly on local industrial devices, reducing latency and enhancing reliability. In large-scale applications, this solution can also be scaled to monitor multiple factories or industrial units across different regions, offering centralized control with decentralized intelligence.

REFERENCES

- Shivam Bansal. *Machine Predictive Maintenance Classification Dataset*. Kaggle.
<https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification>
- IBM. *IBM Watson Studio Documentation*. IBM Cloud Docs.
<https://www.ibm.com/docs/en/watson-studio>
- IBM. *Getting Started with AutoAI in Watson Studio*. IBM Developer Guide.
<https://developer.ibm.com/articles/autoai-watson-studio-overview/>
- IBM Corporation. *AutoAI Algorithms in Watson Studio – Tree Ensemble Classifier (SnapML)*. IBM Cloud Documentation.
<https://www.ibm.com/docs/en/cloud-paks/cp-data/4.5.x?topic=autoai-supported-algorithms>
- IBM. *Deploying machine learning models with Watson Machine Learning*. IBM Cloud Documentation.
<https://www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=models-deploying>
- IBM Cloud. *Watson Machine Learning REST API Documentation*.
<https://cloud.ibm.com/apidocs/machine-learning>

IBM CERTIFICATIONS

GETTING STARTED WITH AI CREDLY CERTIFICATE:



JOURNEY TO CLOUD CREDLY CERTIFICATE:



RAG LAB CERTIFICATE:

IBM SkillsBuild

Completion Certificate



This certificate is presented to
Karunamayudu Doppalapudi

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 24 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU