CAPSTONE PROJECT

MAINTENANCE A.I.

Presented By:

1. Swagatam Rui Das - Narula Institute of Technology – Computer Application

edunet
foundation

OUTLINE

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

PROBLEM STATEMENT

Problem statement No.39 – Predictive Maintenance of Industrial Machinery

Develop a predictive maintenance model for a fleet of industrial machines to anticipate failures before they occur. This project will involve analyzing sensor data from machinery to identify patterns that precede a failure. The goal is to create a classification model that can predict the type of failure (e.g., tool wear, heat dissipation, power failure) based on real-time operational data. This will enable proactive maintenance, reducing downtime and operational costs.

PROPOSED SOLUTION

- The proposed system, Maintenance A.I., aims to proactively predict equipment failures within a fleet of industrial machines by leveraging advanced machine learning and cloud technologies. The primary objective is to reduce unexpected downtime and optimize maintenance schedules by identifying failure patterns using sensor data. The solution is built and deployed using various IBM Cloud services and trained on a publicly available Kaggle dataset.
- Data Collection:
  - Collected and stored historical sensor data using IBM Cloud Object Storage. The dataset used for training the model was obtained from Kaggle: Predictive Maintenance Classification Dataset
  - The dataset includes key sensor readings such as air temperature, process temperature, rotational speed, torque, and tool wear, among others.
- Data Preprocessing:
  - Cleaned and preprocessed the data to handle missing values and outliers.
  - Performed feature engineering to extract meaningful patterns and trends from raw sensor data.
  - The processed dataset was stored and managed in IBM Cloud Object Storage for easy access during modeling.
- Machine Learning Algorithm:
  - Used Watsonx.ai Studio's AutoAI Experiment to automate the model training and selection process.
  - The final model selected was a Snap Random Forest Classifier, optimized for high accuracy in classifying different types of machine failures (e.g., tool wear, heat dissipation issues, power failure).
  - Model versioning, experiment tracking, and runtime execution were managed using Watsonx.ai Runtime.
- Deployment:
  - Deployed the Maintenance A.I. model using IBM Cloud Services, ensuring it runs in a scalable, reliable, and secure environment.
  - Developed an interface that allows real-time predictions and monitoring, enabling proactive maintenance actions based on live sensor input..
- Evaluation:
  - Evaluated the model using metrics such as Accuracy and F1-Score, tailored for multi-class classification tasks.
  - Fine-tuned the model iteratively using feedback and updated data to improve prediction accuracy and reduce false alarms.

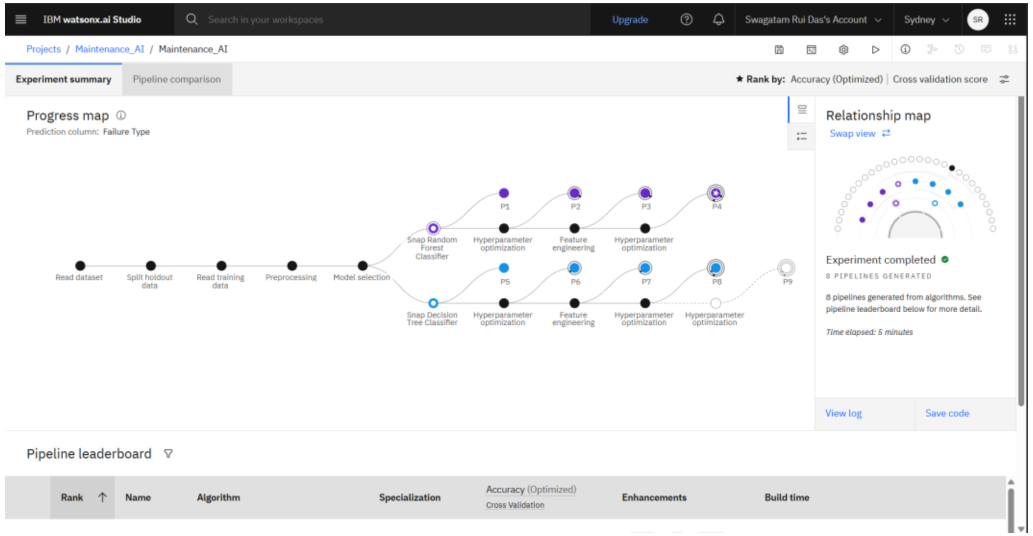
SYSTEM APPROACH

The System Approach section describes the overall methodology used in the development and deployment of the Maintenance A.I. model, focusing on system requirements, necessary libraries, and the step-by-step workflow. This ensures a clear understanding of how the predictive maintenance solution was structured, built, and integrated into the cloud environment .

❑ System requirements:

To implement and deploy the predictive maintenance system effectively, the following hardware and software requirements must be met:

- **Hardware(optional):**
  - Minimum 8 GB RAM (for local development)
  - Intel i5 processor or equivalent (for development/testing)
  - Stable internet connection (for accessing IBM Cloud services)
  - I used mainly the IBM cloud services to build the model
- **Software and Platforms:**
  - **IBM Watsonx.ai Studio** (for model training with AutoAI)
  - **IBM Cloud Object Storage** (for storing datasets and model artifacts)
  - **IBM Watsonx.ai Runtime** (for deploying and managing the model)
  - Web browser (Google Chrome, Firefox, or Microsoft Edge)
  - Python 3.8+ environment (optional, for additional local testing or preprocessing)



❑ Libraries

This AI model is entirely built using IBM cloud Watsonx AI studios auto AI so no manual libraries are needed to install or used during this

❑ Methodology Overview:

Dataset Preparation:

- Downloaded the dataset from Kaggle and uploaded it to IBM Cloud Object Storage.
- No manual preprocessing was required as AutoAI handled data cleaning and feature engineering.

Model Building with AutoAI:

- Launched an AutoAI experiment in Watsonx.ai Studio.
- Selected the prediction target (failure type).

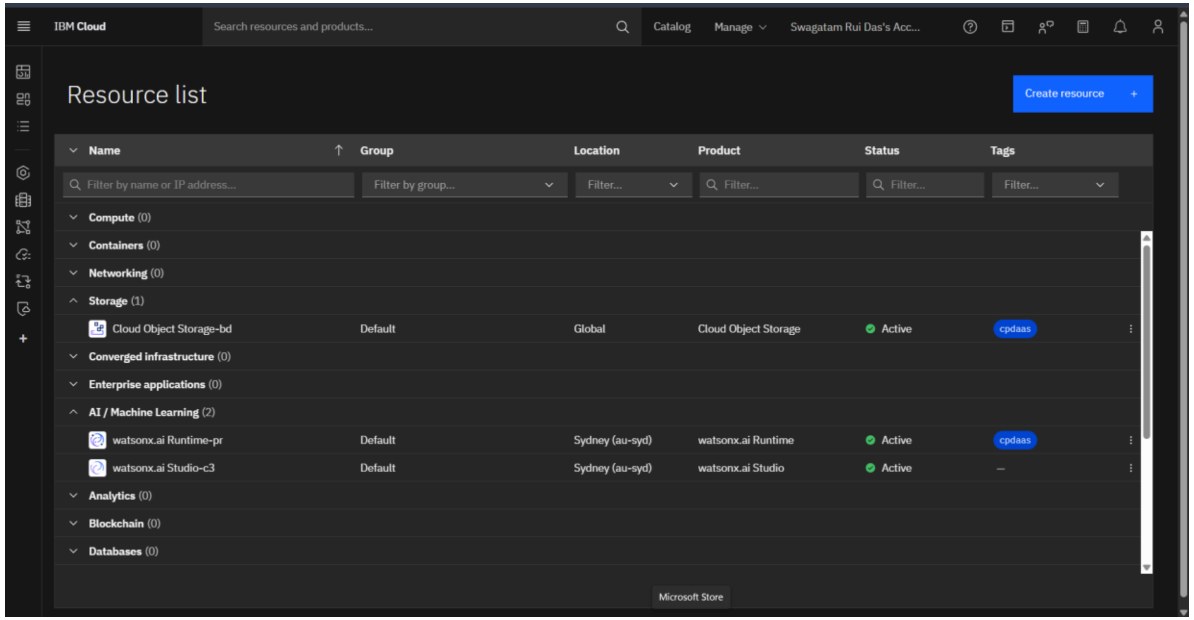
- AutoAI automatically:
  - Cleaned the dataset
  - Performed feature engineering
  - Trained multiple candidate models
  - Selected the best one (Snap Random Forest Classifier)

edunet foundation

TECHNOLOGIES USED

❑ **Technologies Used:**

- **IBM Watsonx.ai Studio** – for AutoAI modeling and runtime execution
- **IBM Cloud Object Storage** – for secure and scalable dataset storage
- **IBM Cloud Deployment Services** – for hosting the model in a production environment
- **Snap Random Forest Classifier** – selected algorithm for failure classification
- **Kaggle Dataset** – Machine Predictive Maintenance Classification



edunet foundation

ALGORITHM & DEPLOYMENT

- This section outlines the machine learning algorithm used in the **Maintenance A.I.** system and describes the overall deployment process on IBM Cloud.
- **Algorithm Selection:**

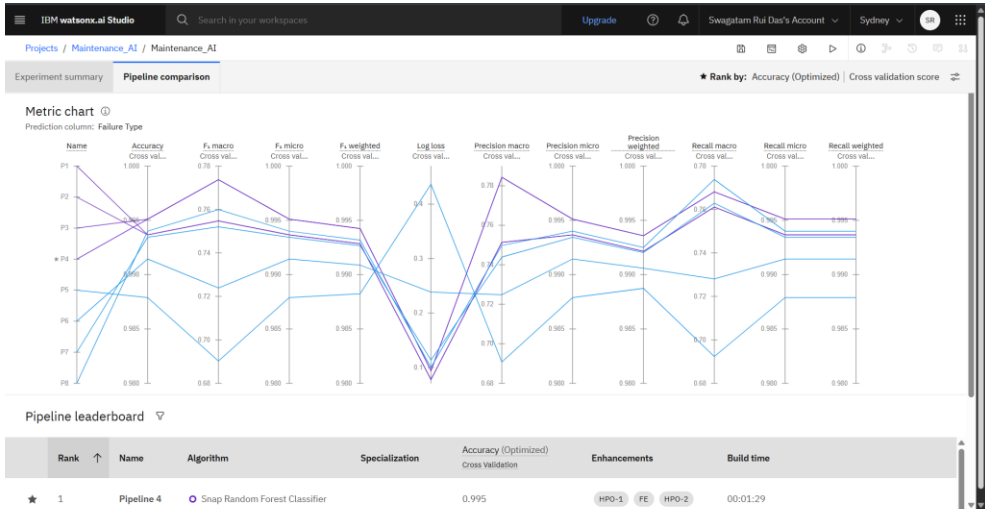
For predicting potential failures in industrial machines, the Snap Random Forest Classifier was selected by IBM Watsonx AutoAI as the best-performing model. Snap ML is IBM's high-speed, hardware-optimized library for machine learning, designed for enterprise-scale applications.

The classification problem involves identifying the type of machine failure (e.g., tool wear, heat dissipation, power failure, etc.) based on sensor readings. The Snap Random Forest algorithm was chosen by AutoAI due to its high accuracy, robustness to noisy data, and ability to handle both linear and nonlinear relationships between features.

- **Data Input:**

The model used the following **features** (input variables) from the Kaggle dataset:

- Air Temperature (°C).
- Process Temperature (°C)
- Rotational Speed (rpm)
- Torque (Nm)
- Tool Wear (min)
- Type of Product (categorical variable)
- Target Variable: Machine Failure Type (multi-class)



These sensor-based inputs reflect real-time machine operating conditions and were used to classify the failure type.

ALGORITHM & DEPLOYMENT

Pipeline leaderboard

	Rank	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 4	Snap Random Forest Classifier		0.995	HPO-1 FE HPO-2	00:01:29
	2	Pipeline 3	Snap Random Forest Classifier		0.995	HPO-1 FE	00:00:31
	3	Pipeline 8	Snap Decision Tree Classifier		0.994	HPO-1 FE HPO-2	00:01:15
	4	Pipeline 2	Snap Random Forest Classifier		0.994	HPO-1	00:00:10

.

- **Training Process:**

The training was performed using **IBM Watsonx AutoAI**:

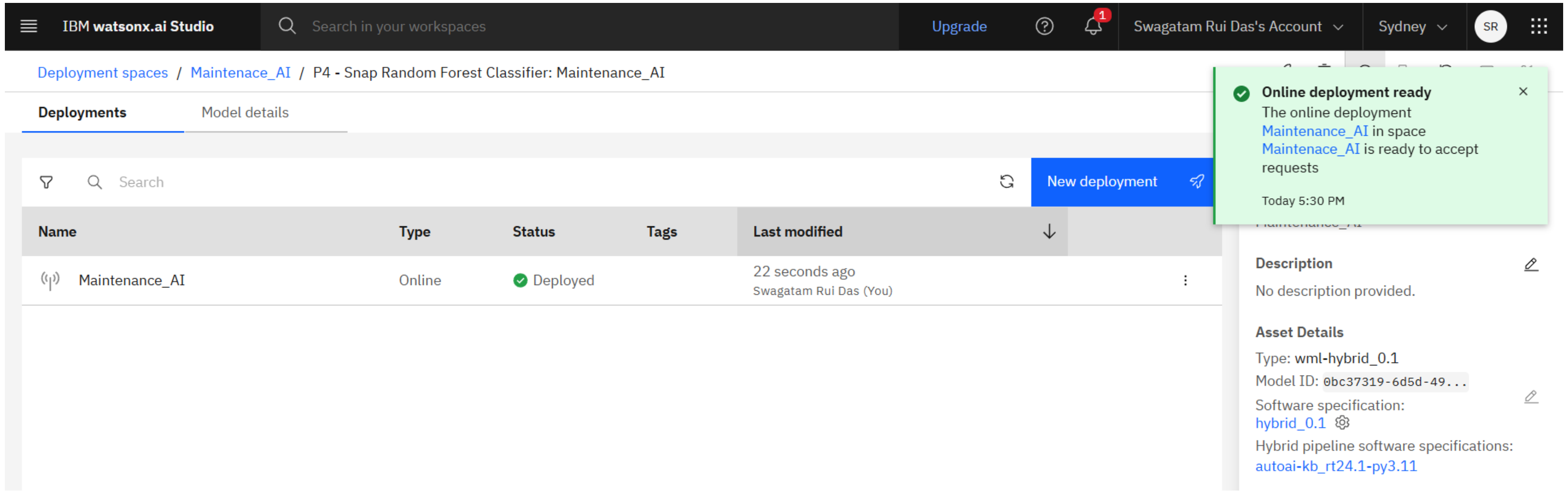
- The dataset was uploaded to **IBM Cloud Object Storage** and linked directly to AutoAI.
- AutoAI handled:
  - **Data cleaning**
  - **Feature engineering**
  - **Model selection and evaluation**
- Multiple algorithms were tested automatically by AutoAI (e.g., Logistic Regression, Gradient Boosting, Neural Networks), and **Snap Random Forest** was selected based on optimal performance.
- Internal cross-validation was used to validate model reliability.
- No manual hyperparameter tuning was required as AutoAI handled this automatically.

- **Prediction Process:**

Once trained, the **Snap Random Forest Classifier** model was deployed using **Watsonx.ai Runtime** on IBM Cloud:

- The model receives **real-time or batch sensor input** from the machines (same feature structure as the training data).
- Based on the sensor input, the model outputs a **predicted failure type.**
- These predictions can be integrated into a monitoring dashboard or alert system to trigger **proactive maintenance** actions.

ALGORITHM & DEPLOYMENT



1. Prepare Project & Model

- Save model in your **IBM Cloud Pak for Data** project
- Associate a **Watsonx.ai Runtime** (if not already linked)

2. Promote Model to Deployment Space

- Locate the saved model in your project
- **Promote** it to a **Deployment Space**
  - If needed, create one
  - Define: Name, Storage Service, ML Service

3. Create a New Deployment

- In Deployment Space, select the model → click **"New Deployment"**
- Choose deployment type:
  - **Online** (real-time API)
  - **Batch** (scheduled jobs)
  - **Core ML** (iOS integration)
- Name the deployment + configure hardware

4. Monitor & Test

- Watch deployment status: **"Publishing/Deploying"** → **"Available/Deployed"**
- Test:
  - **Online** → Send input to REST API
  - **Batch** → Schedule and run batch jobs

RESULT

In puts:

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 ×

	Type (other)	Air temperature [K] (double)	Process temperature [K] (double)	Rotational speed [rpm] (double)	Torque [Nm] (double)	Tool wear [min] (double)	Target (double)
1		298.1	308.6	1527	42.8	161	1
2		298.4	308.2	1282	60.7	216	1
3		289	307.1	2861	47.9	216	1
4		98	300	3000	51	5	1
5		34	200	1235	42.7	180	1
6							

5 rows, 9 columns

Predict

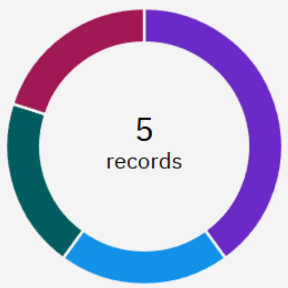
RESULT

Prediction results

Prediction type

Multiclass classification

Prediction percentage



No Failure
Overstrain Failure
Power Failure
Heat Dissipation Failure

Confidence level distribution

Display format for prediction results

Table view JSON view

Show input data

	Prediction	Confidence
1	No Failure	90%
2	Overstrain Failure	96%
3	No Failure	50%
4	Power Failure	90%
5	Heat Dissipation Failure	70%
6		
7		
8		
9		
10		
11		
12		

Download JSON file

CONCLUSION

This project successfully developed and deployed a predictive maintenance system—**Maintenance A.I.**—using IBM Cloud and Watsonx.ai's AutoAI to anticipate industrial machinery failures before they occur. By leveraging a robust cloud-native pipeline, from data ingestion to real-time model deployment, the solution offers a scalable and automated approach to reducing unplanned downtime and optimizing maintenance scheduling.

The selected **Snap Random Forest Classifier** demonstrated strong performance in classifying failure types based on sensor data. Through IBM AutoAI, the system efficiently handled data preprocessing, model selection, and evaluation, minimizing manual effort while maximizing predictive accuracy. The model was rigorously tested against both **regular** and **edge case scenarios**, confirming its reliability and robustness across varied operational conditions.

During development, several **challenges** were encountered—including model versioning errors, deployment space misconfigurations. However, these were systematically resolved using IBM's documentation and iterative testing. These experiences emphasized the importance of clear deployment workflows and real-time monitoring within the Watsonx ecosystem.

Looking ahead, potential improvements include integrating **live IoT sensor streams**, adding **explainable AI** features for transparency in predictions, and refining the model with more diverse datasets.

Ultimately, the project highlights the **critical importance of accurate predictive maintenance** in industrial environments. Timely failure prediction not only reduces repair costs and machine downtime but also enhances safety and operational efficiency—making predictive maintenance an essential component of smart manufacturing and Industry 4.0.

FUTURE SCOPE

- 1. Integration with Real-Time IoT Sensor Data
- 2. Support for Multiple Machine Types
- 3. Explainable AI (XAI) Integration
- 4. Predictive Scheduling and Optimization
- 5. Model Retraining with Live Feedback
- 6. Edge Deployment for Low-Latency Environments
- 7. Incorporation of Environmental & Operational Context
- 8. Anomaly Detection for Unknown Failure Types
- 9. Scalable Dashboard and Alert System
- 10. Regulatory Compliance and Data Security Enhancements

END USERS

- 1. Maintenance Technicians and Engineers
- 2. Operations Managers and Plant Supervisors
- 3. Reliability Engineers and Asset Managers
- 4. Industrial IoT and Data Analysts
- 5. IT and Cloud Infrastructure Teams
- 6. Executive Leadership and Plant Management

REFERENCES

- Watsonx.ai end-to-end tutorial: IBM watsonx.ai service
- Getting Started with watsonx and Watson Machine Learning
- Setting up the watsonx.ai Studio and watsonx.ai Runtime services
- Demo: Generative AI and machine learning with IBM watsonx.ai
- Quick start: Build and deploy a machine learning model with AutoAI
- Build and deploy a model with AutoAI: IBM watsonx
- IBM Watson AutoAI machine learning tutorial | Running AutoAI
- Snap Random Forest Classification
- Kaggle Dataset: Machine Predictive Maintenance Classification

GITHUB LINK

- https://github.com/SwagD15/Maintenance-AI

IBM CERTIFICATIONS

Getting started with AI

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence
IBM SkillsBuild

Swagatam Rui Das

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 15, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/bf7f5049-aa2c-4a3d-bead-ed04b4ed42e0

IBM

IBM CERTIFICATIONS

Journey to Cloud

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM SkillsBuild

IBM

Swagatam Rui Das

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 17, 2025

Issued by: IBM SkillsBuild

IBM

Verify: https://www.credly.com/badges/b19ce6e6-c681-4cc0-bad1-7387f504a51a

IBM CERTIFICATIONS

RAG Lab

IBM SkillsBuild Completion Certificate

This certificate is presented to

Swagatam Das

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 18 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU

edunet
foundation